

eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

# **Machine Learning Approaches to Video Activity Recognition: from Computer Vision to Signal Processing**

Itsaso Rodríguez Moreno

*Supervisors* Basilio Sierra Araujo and José María Martínez Otzeta

January 2023





Konputazio Zientziak eta Adimen Artifiziala Saila  
Informatika Fakultatea

Dissertation submitted in fulfillment of the requirements for the degree of  
Doctor in Computer Sciences

# **Machine Learning Approaches to Video Activity Recognition: from Computer Vision to Signal Processing**

Itsaso Rodríguez Moreno

*Supervisors* Basilio Sierra Araujo and José María Martínez Otzeta

January 2023

**Itsaso Rodríguez Moreno**

*Machine Learning Approaches to Video Activity Recognition: from Computer Vision to Signal Processing*

Dissertation submitted in fulfillment of the requirements for the degree of Doctor in Computer Sciences, January 2023

Supervisors: Basilio Sierra Araujo and José María Martínez Otzeta

**Euskal Herriko Unibertsitatea (UPV/EHU)**

*Robotika & Sistema Autonomoen Ikerketa Taldea (RSAIT)*

Informatika Fakultatea

Konputazio Zientziak eta Adimen Artifiziala Saila

Manuel Lardizabal 1

20018 Donostia-San Sebastián

# Abstract

In recent years Video Activity Recognition has been the main subject of important research efforts due to the importance of its everyday applications, which have also seen their implementation increase by the rapid growth of the associated technologies. Therefore, the demand for automatic recognition of human behavior has grown, making it a highly studied area.

The presented research focuses on classification techniques for two different, although related, tasks, in such a way that the second one could be considered a special case of the first one: recognition of human actions in videos and sign language recognition.

In the first part, the starting hypothesis is that the transformation of the signals in a video using the Common Spatial Patterns algorithm (commonly used in Electroencephalography systems) can produce new features that will be useful for the subsequent video classification by supervised classifiers. Experiments were carried out in several databases, including one created during this research from the point of view of a humanoid robot, with the aim of implementing the developed recognition system in a human-robot interaction setting.

In the second part, the techniques previously developed have been applied to sign language recognition, but, in addition to this task, a method based on the decomposition of the signs is proposed to perform the sign recognition, for a better explainability of the classification process. The final goal is to develop a sign language tutor capable of guiding the users in the learning process, letting them know the mistakes they make and the reason behind them.



# Resumen

En los últimos años el Reconocimiento de Actividades en Vídeo ha sido el objeto principal de una gran cantidad de trabajos de investigación debido a la importancia de sus aplicaciones cotidianas, que también han visto incrementada su implantación por el rápido crecimiento de las tecnologías asociadas. Así, la demanda de reconocimiento automático del comportamiento humano ha crecido, convirtiéndose en un área en la que se desarrolla una gran actividad investigadora.

La investigación presentada se centra en técnicas de clasificación para dos tareas diferentes, aunque relacionadas, de tal forma que la segunda puede ser considerada parte de la primera: el reconocimiento de acciones humanas en vídeos y el reconocimiento de lengua de signos.

En la primera parte, la hipótesis de partida es que la transformación de las señales de un vídeo mediante el algoritmo de Patrones Espaciales Comunes (CSP por sus siglas en inglés, comúnmente utilizado en sistemas de Electroencefalografía) puede dar lugar a nuevas características que serán útiles para la posterior clasificación de los vídeos mediante clasificadores supervisados. Se han realizado diferentes experimentos en varias bases de datos, incluyendo una creada durante esta investigación desde el punto de vista de un robot humanoide, con la intención de implementar el sistema de reconocimiento desarrollado para mejorar la interacción humano-robot.

En la segunda parte, las técnicas desarrolladas anteriormente se han aplicado al reconocimiento de lengua de signos, pero además de ello se propone un método basado en la descomposición de los signos para realizar el reconocimiento de los mismos, añadiendo la posibilidad de una mejor explicabilidad. El objetivo final es desarrollar un tutor de lengua de signos capaz de guiar a los usuarios en el proceso de aprendizaje, dándoles a conocer los errores que cometen y el motivo de dichos errores.





# Laburpena

Azken urteotan, Bideo Ekintzen Sailkapena ikerketa-lan ugariaren helburu nagusia izan da, eguneroko aplikazioetan duen garrantzia dela eta. Izan ere, aplikazio horien ezarpena ere handitu egin da, gaiarekin erlazionatutako teknologia ezberdinek jasan duten hazkunde azkarraren ondorioz. Horrela, giza portaeraren sailkapen automatikoaren eskaria hazi egin da, pisu handiko ikerketa-jarduera eremu bihurtuz.

Ikerketa lan honetan bi ataza desberdinetarako sailkapen-teknikak ikertu dira: giza ekintzen sailkapena bideoetan eta zeinu hizkuntzaren ezagutza automatikoa. Hala ere, bi ataza hauek daukaten harreman estua dela eta, bigarren ataza lehenengoaren zatitzat har daiteke.

Lanaren lehen ataleko hipotesia hurrengoa da: Eredu Espazial Arrunten algoritmoaren bidez (CSP ingelesezko siglen arabera, Elektroentzefalografiako sistemetan erabili ohi dena) bideo baten seinaleak transformatuz, erabilgarriak izan daitezkeen ezaugarri berriak lor daitezke aurrerago sailkatzaile gainbegiratuen bitartez bideo hauen sailkapena egiteko. Esperimentu desberdinak egin dira hainbat datu-baserekin, ikerketa honen testuinguruan robot humanoide baten ikuspuntutik sortutako datu-base bat barne, garatutako sailkapen-sistema robotean ezarriz, gizaki eta roboten arteko elkarrekintza hobetzeko asmoz.

Zeinu hizkuntzaren ezagutza automatikoari dagokionez, aurreko atalean garatutako teknikak erabiltzeaz gain, zeinuen deskonposizioan oinarritutako metodo bat proposatu da zeinu hauek sailkatzeko, sailkapen prozesuaren azalgarritasuna handituz. Azken helburua zeinu hizkuntzarako tutore bat garatzea da, erabiltzaileak ikaskuntza-prozesuan gidatzeko gai dena, egiten dituzten akatsak eta akats horien arrazoiak azalduz.



This research was carried out within the RSAIT group and was supported by the Spanish Ministry of Science, Innovation and Universities, under Grant FPU18/04737.



# Acknowledgements

A thesis is a long journey in which you require guidance, support, and encouragement from many people. I am deeply grateful to those who have helped me along the way.

Now that I am at the end of the journey, I would like to start by thanking those who have helped me from the beginning, my supervisors Basilio Sierra and José María Martínez Ozteta, without your support I would not have known where to start. Thank you Basi for your optimism and for constantly believing in my work. Thank you Ozteta for finding a solution to (almost) all my problems, how reassuring it is to know that you are always just a phone call away.

I would also like to thank each and everyone of the members of the RSAIT group, where I conducted my research, for their assistance, support, friendship and the great time we shared. Igor eta Unai, mila esker egunero nire alboan egoteagatik, denek ez dute zuek bezain "lankide" onak izateko zortea. Izaro, eskerrik asko beti laguntzeko prest egoteagatik, urrutitik izan behar bada ere.

I would like to extend a special thanks to the CAR research group at the University of Galway, where I had the opportunity to conduct a research stay. Thank you for your support from the very first moment and for welcoming me as one of your own.

Gracias a mi cuadrilla porque igual de importante que el trabajo son las tardes de desconexión con ellas. Gracias a los que empezaron este camino conmigo cuando decidí estudiar informática, a los amigos de la uni que se han convertido en compañeros de vida.

Por último, gracias a mi familia, que aunque digan que no entienden mucho lo que hago, siempre sufren mis penas y comparten mis alegrías.

Mila esker! Thank you! Go raibh maith agat! ¡Gracias!



# Contents

<b>I</b>	<b>Research</b>	<b>1</b>
<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Contributions . . . . .	4
1.2.1	Feature extraction . . . . .	4
1.2.2	Action Recognition . . . . .	4
1.2.3	Sign Language Recognition . . . . .	5
1.3	Thesis structure . . . . .	7
<b>2</b>	<b>Theoretical background</b>	<b>9</b>
2.1	Common Spatial Patterns . . . . .	9
2.2	Computer Vision . . . . .	11
2.2.1	Image descriptors . . . . .	11
2.2.2	Pose estimators . . . . .	12
2.2.3	Optical Flow . . . . .	13
2.3	Supervised classification . . . . .	14
<b>3</b>	<b>Action Recognition</b>	<b>17</b>
3.1	Introduction . . . . .	17
3.2	State-of-the-art . . . . .	19
3.3	Contributions . . . . .	20
3.3.1	Datasets . . . . .	20
3.3.2	Preprocessing . . . . .	22
3.3.3	Feature extraction . . . . .	24
3.3.4	Classification . . . . .	25
3.3.5	Results . . . . .	27
3.4	Conclusions and future work . . . . .	29
<b>4</b>	<b>Sign Language Recognition</b>	<b>31</b>
4.1	Introduction . . . . .	31
4.2	State-of-the-art . . . . .	32
4.3	Contributions . . . . .	33
4.3.1	CSP-based approach . . . . .	34
4.3.2	Hierarchical approach . . . . .	36

4.4	Conclusions and future work . . . . .	42
	<b>Bibliography</b>	<b>45</b>
<b>II</b>	<b>Conclusions</b>	<b>51</b>
1	Conclusions and future work	53
<b>III</b>	<b>Publications</b>	<b>55</b>
1	dbcsp: User-friendly R package for Distance-Based Common Spatial Patterns	57
2	Video Activity Recognition: State-of-the-Art	75
3	Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns	103
4	Using Common Spatial Patterns to Select Relevant Pixels for Video Activity Recognition	121
5	A New Approach for Video Action Recognition: CSP-Based Filtering for Video to Image Transformation	139
6	Sign Language Recognition by Means of Common Spatial Patterns	153
7	Towards an Interpretable Spanish Sign Language Recognizer	163
8	A Hierarchical Approach for Spanish Sign Language Recognition: From Weak Classification to Robust Recognition System	173
9	Sign Language Recognition by Means of Common Spatial Patterns: An Analysis	193
10	HAKA: HierArchical Knowledge Acquisition in a Sign Language Tutor	213



# List of Figures

2.1	Body keypoints and hand landmarks extracted with OpenPose and MediaPipe Hands. . . . .	12
3.1	Example of each class of the chosen subset of HMDB51 data. . . . .	21
3.2	Pepper robot and the fields of view of its cameras. . . . .	21
3.3	Example of each class of the dataset recorded by Pepper. . . . .	22
3.4	Process of creating the signals matrix from a video instance . . . . .	23
3.5	Process of creating the signals matrix from skeletons extracted from a video instance . . . . .	23
3.6	Log-variabilities of the projected signals on vectors $\mathbf{a}_1$ , $\mathbf{a}_2$ , $\mathbf{a}_3$ and $\mathbf{b}_1$ , $\mathbf{b}_2$ , $\mathbf{b}_3$ , separated by classes <i>come</i> and <i>five</i> . . . . .	24
3.7	Two different classification approaches based on the images obtained from matrices transformation. . . . .	26
4.1	Example of a frame sequence of a video from the LSA64 database. . . . .	34
4.2	Process of creating the signals matrix from hand landmarks extracted from a video instance . . . . .	35
4.3	Elements which compose the grammar of sign language. . . . .	37
4.4	Process of recognition of configurations . . . . .	38
4.5	Functionalities of the web application . . . . .	40
4.6	Followed pipeline for sign recognition . . . . .	41



# List of Tables

1.1	Summary of the work developed in the presented publications. . . . .	7
2.1	Global visual descriptors used. . . . .	11
3.1	Characteristics of the subset of the HMDB51 dataset used in the approach.	21
3.2	Characteristics of the dataset created with videos recorded by the robot.	22
3.3	Best results obtained for HMDB51 subset dataset, obtained by Random Forest classifier. . . . .	27
3.4	Best accuracy values obtained with the proposed methods, where the results of the best method of each type of approach are presented. . . .	29
4.1	Characteristics of the LAS64 dataset: one-handed signs. . . . .	34
4.2	Configuration of the parameters used in the classification process. . . .	36
4.3	Obtained results for LSA64 dataset with different parameter values. . .	36
4.4	Definition of the selected signs and number of instances used to create the databases. . . . .	41
4.5	Regular expressions corresponding to each of the selected signs. . . .	42



# Part I

---

Research



# Introduction

In recent years, due to its multiple beneficial applications, different techniques have been developed to analyse visual content and obtain information from it in order to apply it in different areas such as autonomous driving, video surveillance or human-robot interaction. The concept of using machines to understand and analyse visual content, such as images and videos, pertains to the field of computer vision.

These algorithms have evolved over several decades but recent advances in machine learning techniques, as well as improvements in data storage, computing capabilities and low-cost high-quality devices, have driven significant improvements in analysing visual content.

Within computer vision, this research focuses on video action recognition and sign language recognition, presenting several approaches that make use of different machine learning techniques.

## 1.1 Motivation

Automatic video action recognition offers many real-world applications, being human-computer interaction one of them. In the Robotics and Autonomous Systems Research Group (RSAT, in Basque) where this thesis dissertation has been carried out, Social Robotics is one of the fields worked on; endowing the robot with the ability to recognise different human actions and to be able to react to them in an appropriate way offers a wide range of possibilities for improvement when it comes to interacting with users. To this end, different approaches are presented for the recognition of a set of actions.

Sign language recognition can be considered a subfield of action recognition. Currently, more than 5% of the world's population suffers from disabling hearing loss. The consequences for hard of hearing people in an environment without an interpreter can be severe, including loss of the ability to communicate with others, which can result in feeling lonely, frustrated, and socially isolated. Although there are more than 300 different sign languages, they are not widespread among the hearing population. They are commonly known by hard of hearing people, their relatives

and professionals. Therefore, developing a sign language recognition system would be very useful for different areas of everyday life. In addition, a tutor to help users who are learning a sign language can be used to show them the mistakes they are making when performing the sign and thus assist them in the corrections.

## 1.2 Contributions

The work carried out in this research has resulted in a series of contributions related to these different fields: feature extraction, action recognition and sign language recognition.

### 1.2.1 Feature extraction

- [Rod+22] Itsaso Rodríguez et al. "dbcsp: User-friendly R package for Distance-Based Common Spatial Patterns". In *The R Journal* 14.3 (2022), pp. 80-94.

In this paper the original Common Spatial Patterns method is extended. In mathematical terms, Common Spatial Patterns (CSP) is based on the generalized eigenvalue decomposition or the simultaneous diagonalization of two matrices to find projections in a low dimensional space. The original method is based on the Euclidean distance between signals, and here an extension is proposed so that it can be applied on any appropriate distance for data at hand. Both the classical CSP and the new Distance-Based CSP (DB-CSP) are implemented in an R package, called *dbcsp*.

### 1.2.2 Action Recognition

- [RM+19] Itsaso Rodríguez-Moreno et al. "Video activity recognition: State-of-the-Art". In: *Sensors* 19.14 (2019), p. 3160.

The aim of this paper is to survey the state-of-the-art techniques for video activity recognition while at the same time mentioning other techniques used for the same task that the research community has known for several years. For each of the analysed methods, its contribution over previous works and the proposed approach performance are discussed.

- [RM+20a] Itsaso Rodríguez-Moreno et al. "Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns". In: *Sensors* 20.8 (2020), p. 2436.



In this work, a video activity recognition method is presented, which has the ultimate goal of endowing a robot with action recognition capabilities for a more natural social interaction. The application of CSP is presented in a novel manner to be used in activity recognition in videos taken by a humanoid robot. A sequence of skeleton data is considered as a multidimensional signal and filtered according to the CSP algorithm. Then, characteristics extracted from these filtered data are used as features for a classifier.

- [RM+20b] Itsaso Rodríguez-Moreno et al. "Using Common Spatial Patterns to Select Relevant Pixels for Video Activity Recognition". In: *Applied Sciences* 10.22 (2020), p. 8075.

In this paper, taking as a basis the work of [RM+20a], a new approach for video action recognition is presented, where input videos are represented as frame sequences and the temporal sequence of each pixel is treated as a signal (channel) to feed the CSP. After CSP is applied, some signals descriptors are selected for classification purposes.

- [RM+21a] Itsaso Rodríguez-Moreno et al. "A New Approach for Video Action Recognition: CSP-Based Filtering for Video to Image Transformation". In: *IEEE Access* 9 (2021), pp. 139946–139957.

In this paper, the CSP algorithm is applied to a set of signals obtained for each video by extracting skeleton joints of the person performing the action. From the transformed signals a summary image is obtained for each video, and these images are then classified using two different approaches; global visual descriptors and Convolutional Neural Networks (CNN).

### 1.2.3 Sign Language Recognition

- [RM+21b] Itsaso Rodríguez-Moreno et al. "Sign Language Recognition by Means of Common Spatial Patterns". In: *2021 The 5th International Conference on Machine Learning and Soft Computing*. 2021, pp. 96–102.

In this work, an Argentinian Sign Language (LSA) recognition system is presented which distinguishes between different signs using hand landmarks extracted from the videos of the dataset. The CSP algorithm is used to extract features, and the classification is performed with multiple classifiers.

- [RM+22b] Itsaso Rodríguez-Moreno et al. "Towards an Interpretable Spanish Sign Language Recognizer". In: *ICPRAM*. 2022, pp. 622–629.

In this paper a blueprint for a sign language recogniser that takes advantage of the internal structure of the signs of the Spanish Sign Language (SSL) is presented. The signs are decomposed into constituents which are in turn recognised by a classical classifier and then assessed if their combination is congruent with a regular expression associated with a whole sign. While the deep learning with many examples approach works for every possible collection of signs, this approach could leverage the known structure of the sign language in order to create simpler and more interpretable classifiers that could offer a good trade-off between accuracy and interpretability.

- [RMMOS22a] Itsaso Rodríguez-Moreno et al. "A Hierarchical Approach for Spanish Sign Language Recognition: From Weak Classification to Robust Recognition System". In: *Proceedings of SAI Intelligent Systems Conference*. Springer. 2022, pp. 37–53.

This paper presents the sign language recognition module from an ongoing effort to develop a real-time Spanish sign language recognition system that could also work as a tutor. The proposed approach focuses on the definitions of the signs, first performing the classification of their constituents to end up recognising full signs.

- [RM+22a] Itsaso Rodríguez-Moreno et al. "Sign Language Recognition by Means of Common Spatial Patterns: An Analysis". In: *Plos one* 17.10 (2022), e0276941.

In this work, an Argentinian Sign Language recognition system is presented. It uses hand landmarks extracted from videos of the LSA64 dataset in order to distinguish between different signs. Different features are extracted from the signals created with the hand landmarks values, which are first transformed by the CSP algorithm. The features extracted from the transformed signals have been then used to feed different classifiers, such as Random Forest (RF), K-Nearest Neighbors (KNN) or Multilayer Perceptron (MLP).

- [RMMOS22b] Itsaso Rodríguez-Moreno et al. "HAKA: HierArchical Knowledge Acquisition in a Sign Language Tutor". In: *Expert Systems with Applications journal* 215 (2022), p. 119365.

In this paper, a tutor has been developed for learning the basic 42 hand configurations of the Spanish Sign Language, as well as more than one hundred of common words. This tutor registers the user image from an off-the-shelf webcam and challenges her to perform the hand configuration she chooses to practice. The system looks for the configuration, out of the 42 in its database,

closest to the configuration performed by the user, and shows it to her, to help her to improve through knowledge of her errors in real time. The similarities between configurations are computed using Procrustes analysis. A table with the most frequent mistakes is also recorded and available to the user. The user may advance to choose a word and practice the hand configurations needed for that word.

A summary of the methodologies used in each approach is presented in Tab. 1.1.

**Tab. 1.1:** Summary of the work developed in the presented publications.

Article	Summary
[Rod+22]	Implementation of distance-based CSP algorithm in R available on CRAN.
[RM+19]	Review of state-of-the-art methods for Video Activity Recognition task.
[RM+20a]	Skeleton extraction + CSP projection + feature extraction and classification.
[RM+20b]	Relevant pixel selection + CSP projection + feature extraction and classification.
[RM+21a]	Keypoints extraction + CSP projection + matrix transformation + image classification (CNN global visual descriptors).
[RM+21b]	Hand landmark extraction + CSP projection + feature extraction and classification.
[RM+22b]	Hand landmark extraction + hand shape (configuration) learning + Regular expressions.
[RMMOS22a]	Hand landmark extraction + hand shape (configuration) learning + Hidden Markov Models.
[RM+22a]	Hand landmark extraction + CSP projection + feature extraction and classification.
[RMMOS22b]	Hand landmark extraction + create ground-truth models dataset + Procrustes disparity + web application.

## 1.3 Thesis structure

This PhD dissertation is divided into three main parts. The contents are organized as follows.

**Part I:** The first part is focused on the research carried out and is divided into four chapters.

*Chapter 1:* The first chapter is an introduction that contains the motivation and contributions of the presented work.

*Chapter 2:* This chapter focuses on the theoretical backgrounds of the principal techniques used in the development of the presented research. In addition to several machine learning and deep learning techniques, the Common Spatial Patterns method is presented, which has been used for transforming the input signals and obtaining new features to be used in the recognition process.

*Chapter 3:* The third chapter focuses on the action recognition task, including the contributions made in this area. The main idea behind the proposed approaches for action recognition is to use the Common Spatial Patterns as a feature extraction method. The signals to feed the CSP algorithm are obtained in two different ways: (1) sequences of pixels or (2) sequences of skeletons (sets of joints) extracted from people performing the actions.

*Chapter 4:* Chapter 4 is centered on Sign Language Recognition. Although the idea presented for general action recognition has also been tested for sign language recognition, the contributions made in this area focus on explainability and the ability to interpret the obtained results in order to develop a tutor for learning Spanish Sign Language.

**Part II:** Conclusions are presented in Part II, along with ideas related to the work to be done in the future.

**Part III:** Finally, the publications supporting the presented contributions are collected in Part III.

## Theoretical background

Throughout the research carried out, different techniques have been used for feature extraction as well as for classification and computer vision. In the following section the diverse methods used and the underlying theory behind them are explained.

### 2.1 Common Spatial Patterns

The Common Spatial Patterns (CSP) algorithm is a feature extraction method that makes use of optimum spatial filters to process signals of multiple channels. It was first proposed under the name Fukunaga-Koontz Transform [FK70] as an extension of Principal Components Analysis (PCA). The CSP algorithm is based on the generalized eigenvalue decomposition or the simultaneous diagonalization of two covariance matrices to find projections in a low-dimensional space. Specifically, the algorithm is applied on data corresponding to two different classes, thus being a supervised technique, and each individual is represented by several signals recorded during a time period. The aim of the CSP is to find the optimal spatial filters that maximize the variance of the signals of one class while minimizing the variance of the signals of the other class, thus attempting to discriminate the two classes in the best possible way.

Let us consider  $C_1$  and  $C_2$  classes with  $n_1$  and  $n_2$  number of instances, respectively. Each individual is represented by a matrix  $X_{ik}$  where  $i = 1 \dots n_k$  represents an instance and  $k = 1, 2$  the corresponding class. Each  $X_{ik}$  is a matrix of  $c \times T$  dimension, representing the  $c$  sources or signals collected during  $T$  time units.

The feature extraction by means of CSP is performed as follows:

- Compute the average covariance matrices of both classes:

$$B_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} X_{i1} X'_{i1}, \quad B_2 = \frac{1}{n_2} \sum_{i=1}^{n_2} X_{i2} X'_{i2}$$

- Compute the generalized eigendecomposition of the covariance matrices

$$B_1 \mathbf{w} = \lambda B_2 \mathbf{w}$$

to find the directions  $W = (\mathbf{w}_1, \dots, \mathbf{w}_c) \in \mathbb{R}^{c \times c}$  according to:

$$\begin{aligned} & \text{Maximize } tr(W' B_1 W) \\ & \text{subject to } W'(B_1 + B_2)W = I \end{aligned}$$

- Choose the first and last  $q$  vectors  $W_{CSP} = (\mathbf{w}_1, \dots, \mathbf{w}_q, \mathbf{w}_{c-q+1}, \dots, \mathbf{w}_c)$ . The new signals  $Z$  are obtained by multiplying the spatial filter  $W_{CSP}$  with the original signals  $X_{ik}$ .

$$Z = W'_{CSP} X_{ik}$$

This way, the first  $q$  vectors ( $j = 1, \dots, q$ ) obtain large variability for signals belonging to class  $C_1$  ( $X'_{i1} \mathbf{w}_j$ ) and low variability for signals belonging to class  $C_2$  ( $X'_{i2} \mathbf{w}_j$ ), and the opposite is obtained with the last  $q$  vectors ( $j = c - q + 1, \dots, c$ ).

- Create the feature vector for classification purposes.

$$f_p^i = \log \left( \frac{var_p(Z_i)}{\sum_{p=1}^{2q} var_p(Z_i)} \right)$$

where  $var_p(Z_i)$  is the variance of the signal  $p$  of the  $i$ -th instance of  $Z$ . A feature vector of  $2q$  dimensionality is obtained, where  $q$  indicates how many vectors of the spatial filter have been used in the projection.

Based on the classical CSP method, which uses the Euclidean distance between the signals, a generalization of this method called Distance Based Common Spatial Patterns (DB-CSP) which offers the option to use any distance has been developed in a work coauthored by the author of this thesis. Therefore, the covariance matrices are calculated as follows to take into account the chosen distance.

$$B_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \left( P_{ik} P'_{ik} + X_{ik} \mathbf{x}_{ik} \mathbf{1}' + \mathbf{1} \mathbf{x}'_{i,k} X'_{ik} - \mathbf{x}'_{i,k} \mathbf{x}_{ik} \mathbf{1} \mathbf{1}' \right)$$

where  $\mathbf{x}_{ik} = \frac{1}{c} \mathbf{1}' \mathbf{X}_{ik}$  and

$$P_{ik} = -1/2 H D_{ik}^{(2)} H$$

where  $D_{ik}$  stands for the distance matrix of the selected distance,  $H$  for the centering matrix and (2) for squared elements in the matrix.

The classical CSP has been used in the research as a feature extraction method, which will be further explained in the following sections. On the other hand, an R package has been built for the presented extension called DB-CSP which can be found on CRAN <sup>1</sup> and it is further explained in the following publication:

- **dbcsp: User-friendly R package for Distance-Based Common Spatial Patterns.** Itsaso Rodriguez, Itziar Irigoien, Basilio Sierra, and Concepcion Arenas. In *The R Journal* 14.3 (2022), pp. 80-94.

## 2.2 Computer Vision

Computer vision focuses on extracting information from images and videos, involving techniques used to process, analyse and understand these images. To be able to both detect an action being performed in a video and to perform sign language recognition, it is necessary to apply different computer vision techniques to extract information from the images that have been (or are being) recorded in order to process them and classify what is happening in the image/video. Next, the techniques used throughout the study are presented, including image descriptors, pose estimators and optical flow.

### 2.2.1 Image descriptors

Different information can be extracted from an image with the use of image descriptors. These descriptors describe visual features of images or videos, encoding into a list of numbers some information of interest for classification purposes. These visual characteristics can refer to basic features such as shape, color, texture or motion. The different visual descriptors used in this research and the information they offer are indicated in Tab 2.1.

**Tab. 2.1:** Global visual descriptors used.

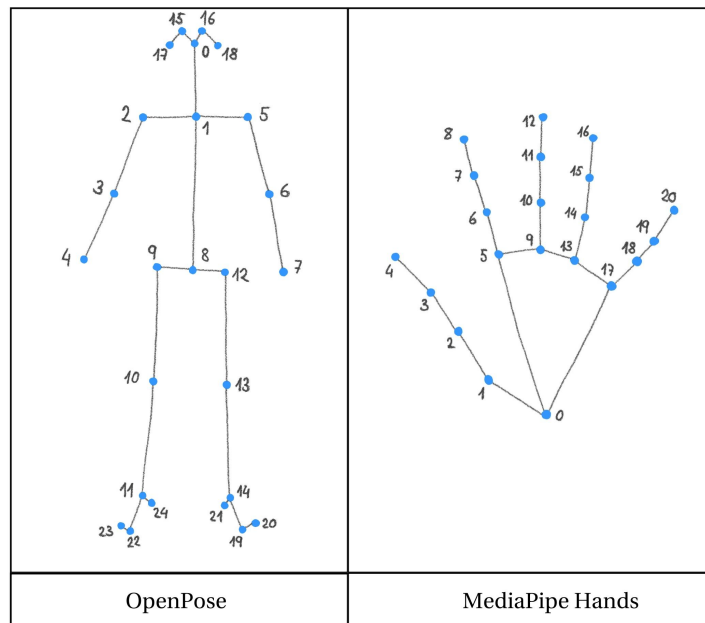
Descriptor name	Information obtained from an image
Color Layout Descriptor (CLD)	Spatial distribution of color
Pyramid Histogram of Oriented Gradients (PHOG)	Local shape and the spatial information of the shape
Fuzzy Color and Texture Histogram (FCTH)	Histogram joining color and texture information
Edge Histogram Descriptor (EHD)	Histogram of the edges directions

<sup>1</sup><https://CRAN.R-project.org/package=dbcsp>

## 2.2.2 Pose estimators

Since for action recognition it could be of great interest to consider the position of the actor (the person performing the action) and his or her movements, it is important to obtain the pose estimation of the actor. Pose estimation refers to capturing the location of human joints in order to use this information to be able to analyse body movement. Two different pose estimators have been used (OpenPose and MediaPipe Hands), which extract the keypoints displayed in Fig 2.1.

- OpenPose [Cao+17]: it can detect human body, feet, hands, and facial keypoints (135 keypoints in total) on single images. However, the BODY\_25 (COCO [Lin+14] + feet) model has been used for human pose estimation, which returns the  $(x, y)$  positions of 25-keypoints, including head, body, and feet.
- MediaPipe [Lug+19]: although the MediaPipe suite also offers body keypoints, the MediaPipe Hands [Zha+20] utility is used to extract hand landmarks for sign language recognition purposes. It estimates  $(x, y, z)$  positions of 21 hand landmarks for each hand, where  $x$  and  $y$  are the horizontal and vertical position in the image normalized to  $[0.0, 1.0]$ , and  $z$  refers to the relative depth with reference to the wrist and it is given roughly in the same scale as  $x$ .



**Fig. 2.1:** Body keypoints and hand landmarks extracted with OpenPose and MediaPipe Hands.



### 2.2.3 Optical Flow

When dealing with videos, the scene changes from frame to frame, so techniques capable of describing the movement of the image are needed. Optical flow is a technique suited to analyse the motion in a video sequence. Specifically, the optical flow quantifies the position change of objects in consecutive frames, which could be caused by an intrinsic movement of the object being captured or by the movement of the camera that captures it.

Therefore, optical flow methods attempt to compute the motion of the objects shown in two consecutive images captured at time  $t$  and  $t + \Delta t$ .

The grayscale value of a pixel can be represented as  $I(x, y, t)$ , where  $I$  refers to the intensity as a function of three variables, where  $(x, y)$  are the spatial coordinates and  $t$  is the time. If the pixels move  $(\Delta x, \Delta y)$  in the next elapsed frame at time  $\Delta t$ , and assuming that the intensity does not vary during the movement, the following equation is obtained:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

Developing the constraint with the Taylor Series Approximation we obtain the following:

$$\begin{aligned} I(x + \Delta x, y + \Delta y, t + \Delta t) &= I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t + \dots \\ \implies \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t &= 0 \end{aligned}$$

After dividing by  $\Delta t$ :

$$\begin{aligned} \frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} \frac{\Delta t}{\Delta t} &= 0 \\ \frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} &= 0 \end{aligned}$$

The gradients  $\frac{\partial I}{\partial x}$ ,  $\frac{\partial I}{\partial y}$ ,  $\frac{\partial I}{\partial t}$  of  $(x, y, t)$  are known but the components of the optical flow (velocity)  $V_x$  and  $V_y$  have to be estimated and it can not be done with just an equation. There are several methods that include more restrictions to be able to compute the optical flow.

While sparse optical flow applies different algorithms to obtain the flow of just some points of an image considered of special interest, dense optical flow computes the optical flow for every point of the frame, obtaining one velocity vector for each pixel.

## 2.3 Supervised classification

The aim of artificial intelligence is to automate processes that, if performed by humans, would be considered product of cognitive processes.

Machine learning is the subfield of artificial intelligence focused on achieving that goal through the analysis of data with different techniques that stem mainly from statistics. Those techniques are adapted to problems that do not admit a closed solution and usually take the form of an iterative loop over a training set in which an initial model is refined through the minimization of some penalty function.

Machine learning can either be supervised or unsupervised. In supervised learning the training cases are labeled, i.e., each case is related to its class. On the contrary, in the unsupervised learning setup, there is no class information related to the instances.

When a new unlabelled instance arrives, a classifier can be used to make a prediction of a class for that instance, i.e. to assign a label to that new instance. There are several algorithms to obtain this classifier, which use information obtained from previously known data (training database) to perform the classification. Therefore, these algorithms try to define, from previously known information, a classification rule for classifying new cases. The present research is focused on supervised learning since the training databases used are labeled and the goal is to predict the labels on new unlabeled instances.

There are different types of classifiers and some of them are briefly defined next.

**Linear Discriminant Analysis (LDA).** It tries to separate the different classes by finding a linear projection that maximizes the distance between classes and minimizes the distance between the objects of the same class.

**Support Vector Machines (SVM).** It maps the training cases to a space of a larger dimension, where it is possible to find a hyperplane that better separates them into classes, and classifies the new cases according to their location with respect to this hyperplane.

**K-Nearest Neighbors (KNN).** As the name suggests, the new instance is assigned the class that is most repeated among its K nearest neighbours.

**Naive Bayes (NB).** It is a classifier based on Bayes theorem. In this case, the posterior probability for each value of the class variable given the data is calculated with the information from the predictor variables, assuming that there is no dependence between the predictor variables. The most likely class is predicted.

**Bayesian Network (BN).** It is a probabilistic graphical model, based on the Bayes theorem, which provides a representation of a set of variables and the dependencies between them. The structure is a directed acyclic graph where each node refers to a unique variable and the edges represent the conditional dependency amongst them.

**Decision trees.** Set of constraints organized in a hierarchical structure. Given a database, different attribute-based paths are established until a class is predicted.

- **J48:** this classification tree is constructed using the C4.5 algorithm; if all the instances in the set are of the same class this is placed as a leaf, otherwise a sub-tree must be formed with the variable that satisfies  $X = \max \frac{I(X, C)}{H(X)}$  condition, where  $I(X, C)$  refers to the mutual information and  $H(X)$  to the entropy.
- **Random Forest (RF):** a Bagging<sup>2</sup> (Bootstrap Aggregating) multi-classifier that trains a group of decision trees (forest) based on different subsets of randomly selected predictor variables for training.

**Hidden Markov Models (HMM).** A probabilistic graphical model where a sequence of unknown (hidden) states can be predicted from a set of observations, that is, a Markov model where the states are not directly observable. Let  $s_t$  and  $o_t$  be the hidden state and the observation for each time  $t$ , respectively. On the one hand, in each step  $t$  the actual state just depends on the previous state,  $\mathbf{P}(s_t | s_1, s_2, \dots, s_{t-1}) = \mathbf{P}(s_t | s_{t-1})$ , which is known as the Markov property. On the other hand, the observation just depends on the actual state  $\mathbf{P}(o_t | s_1, \dots, s_t, o_1, \dots, o_{t-1}) = \mathbf{P}(o_t | s_t)$ . An HMM is defined by (1) the transition data (the probability  $\mathbf{P}(s_j | s_i)$  of transition from state  $s_i$  to state  $s_j$ ), (2) the emission data (the probability  $\mathbf{P}(o_t | s_t)$  of emitting the observation  $o_t$  from state  $s_t$ ), and (3) state prior probability (the initial probability of starting in state  $s_i$ ).

---

<sup>2</sup>**Bagging:** the combination of several classifiers of the same type trained with different subsets of the database.

In recent years, thanks to computational advances and improvements in algorithms, the use of deep learning in recognition and classification problems has exploded. When training classifiers, the features to be used to train these models must be obtained. While machine learning algorithms require manual feature selection, when using deep learning algorithms, the algorithm could automatically retrieve the features that are important for learning. Automatic feature selection is a great advantage, as feature extraction and selection is a time-consuming and difficult process.

This research focuses on video recognition, so there are two main aspects to take into account: the spatial analysis of the image constituents and the importance of temporal information. Taking these features into consideration, two different deep learning techniques have been used throughout the study.

**Long Short-Term Memory (LSTM).** LSTMs are a category of recurrent neural networks (RNNs) which possess an internal state that stores information about past inputs, which endows them with the ability to process sequences of inputs. LSTM design introduces gates that control how much of the past and the current state has to get through to the next time step. There are three gates: (1) *forget gate* removes useless information, (2) *input gate* adds useful information, and (3) *output gate* gives an output extracting the useful information from the current cell.

**Convolutional Neural Network (CNN).** CNNs are neural networks focused on image analysis and classification. Feature extraction is performed by convolutional layers which apply mathematical operations to groups of pixels using a convolution matrix, the kernel. After each convolutional layer a downsampling layer is applied, the pooling layer, and at the end of the network simpler perceptron neurons are used to perform the final classification of the extracted features.

# Action Recognition

## 3.1 Introduction

Video action recognition is a task that involves analysing and recognising the action (or actions) that one or more actors are performing in a series of observations. Thus, the goal is to be able to automatically detect the actions that are being performed in a video.

As technology develops quickly, so does the demand for automated analysis of human behavior in videos, making video action identification a very active field. Therefore, due to the growth of the availability of multimedia files and the variety of applications it may be used for, action recognition has seen an increase in interest over the past several years and as large volumes of data are currently generated by multimedia information, the necessity for developing automatic or semi-automatic systems that enable the labeling of video actions for various applications increases.

Action Recognition has various real-life applications, including visual surveillance, rehabilitation, human-computer interaction, and entertainment, and it is mostly a subfield of computer vision since the visual elements give fundamental information about what is happening in the image sequence. For example, since manual tagging has become tedious and occasionally impractical due to the enormous growth that multimedia data has experienced in recent years, video recognition can be used to perform annotation and indexing of videos. Another application is the use of action recognition in rehabilitation, which requires recognising the activity patients are performing and being able to tell whether they are doing it correctly or not. Regarding surveillance, a system that gives real-time alerts concerning suspicious activity is quite helpful in a variety of circumstances. This work can also be helpful in creating interactive systems for the entertainment sector or for social good, such as aiding people with some impairment.

Humans can analyse and interpret visual data quickly and easily, without being very affected by environmental adversities such as occlusions, illumination, etc. However, although throughout the years many different approaches have been presented trying to identify actions in videos, this is a task not so easy to automatically perform. Challenges such as image classification and face recognition share common

characteristics with the task of video activity recognition, but due to several factors the latter has not progressed in the same way despite the efforts made by the research community.

A video is composed of a sequence of images that occur in a certain order, therefore, the temporal dimension, as well as the spatial features, must be taken into account when analysing them. That need of combining both spatial and temporal information adds extra complexity to the task. Among the difficulties of this task are the following:

- **The temporal information is crucial.** For example, if the frames of a video were analysed independently, there would be no difference between a frame corresponding to a person walking and another corresponding to a person running, since the difference between these two actions is centered on the speed at which the action is performed. Hence, more than a single frame is needed to determine the action that is taking place.
- **Intra-class variations.** It exists a high intra-class variability between the instances, caused by different factors. These factors range from the diversity of actors to camera movements or even changes in the environment. For instance, the appearance of the actors and the way they execute the actions vary, the environment can be affected by changes in lighting, shadows and occlusions, the distance from the subject to the camera may differ and there can be differences in the resolution of the videos. For all these details two instances corresponding to the same action are almost always far from identical.
- **Resolution and computational cost.** A balance must be found between video resolution and computational cost, because if the resolution is too low, visual information may be lost, but if the resolution is too high, greater computational resources are also required. The same must be decided with the frame rate (fps) to avoid losing a significant amount of temporal information without requiring a large computational capacity.
- **Limited datasets.** Due to the difficulty of collecting, labeling and storing large volumes of data, it is a challenge to find complete and suitable video databases to perform this task.

## 3.2 State-of-the-art

Several approaches have been developed over the years to address the problem of action recognition in video [Bed+20]. These techniques make use of both static and temporal visual features extracted from the video, which can be divided into different groups: the identification of spatio-temporal points of interest, the representation of the action sequence as a 3D spatio-temporal volume, the use of skeleton information of the person performing the action, and the use of deep learning to process sequences of frames.

Space-time interest points extracted from video have been widely used for action recognition. The authors of [NYV18] use 3D Harris Space-Time Interest Point detector and 3D Scale-Invariant Feature Transform descriptor for feature extraction. Finally, the videos are compactly represented with a histogram of visual features using the bag-of-visual feature approach. In [NFF07] the authors present a hybrid hierarchical model, where video sequences are represented as collections of spatial and spatio-temporal features that are obtained by extracting both static and dynamic interest points. Static features are obtained computing an edge map using Canny edge detector and motion features using a separable linear filter method. Many other methods make use of Histograms of Oriented Gradients (HOG) or Histogram of Oriented Optical Flow (HOOF) [TS08; CA09; Cha+09]. Motion descriptors based on the direction of optical flow have also been introduced [LAC11].

Due to advances in imaging technology to capture real-time depth information, there are already many approaches that use this extra information to try to improve activity recognition. Using RGB depth cameras (RGB-D), which are robust to illumination changes, depth maps that provide information about the distance of objects from a viewpoint are obtained and used to perform recognition [STC18; LLC17; Ari+19b; Wan+15].

The information of the skeleton of people performing the action to recognise has also been used in many studies. For instance, the authors of [Cho+18] represent videos by means of the movement of human joints, using a pose estimator to extract joint heatmaps for each frame and colorizing them depending on the relative time in the video clip. In [Shi+19] a two-stream adaptive graph convolutional network (2s-AGCN) is proposed, where the coordinates of the joints and the bones between the joints are used as features for classification along both the spatial and temporal dimensions.

Due to their performance in image recognition, deep learning methods have recently been applied to video-based activity recognition [HHP17; YLZ19], where mainly

Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs) have been used. The work presented in [Ull+17] proposes the use of CNNs and deep bidirectional LSTM (DB-LSTM) for the recognition, while the authors of [Ari+19a] obtain a final motion map of the whole video by iteratively extracting motion maps with a deep 3-dimensional CNN and use an LSTM for the final prediction. In [DLL20] a two-stream attention-based LSTM is proposed, which consists of a temporal feature stream focused on the optical flow and a spatio-temporal feature stream that focuses on the effective features and assigns different weights to the outputs of each deep feature map. Authors of [LGH19] propose a Temporal Shift Module (TSM) that shifts the channels forward or backward along the temporal dimension to exchange information between adjacent frames.

## 3.3 Contributions

The approaches we present throughout this document are based on the use of the CSP algorithm as a feature extractor for Action Recognition in videos. As explained in Section 2.1, the CSP algorithm is applied to two lists of matrices of signals corresponding to two different classes and tries to find an optimal filter that separates the two classes according to the variance of these signals.

The recognition process can be divided into different parts, which are explained below, along with the datasets used.

### 3.3.1 Datasets

#### **A — Subset of HMDB51 dataset**

Although there are several databases focused on action recognition [KF22], for a first approximation it was decided to use a subset of the HMDB51 database [Kue+11], which contains videos from movies as well as from Youtube, Google or Prelinger Archives. As the videos are extracted mainly from movies, the dataset is really challenging, due to the variety of points of view, camera movements and background. Therefore, the six classes shown in Fig 3.1 have been selected to be used in our approach, where a frame of an example video is shown for each chosen class.

The main features of the database can be seen in Tab. 3.1.

#### **B — Dataset recorded with Pepper humanoid robot**

In order to enhance human-machine interaction and integrate robots into intricate human situations, robots must comprehend and adapt to human behavior in order





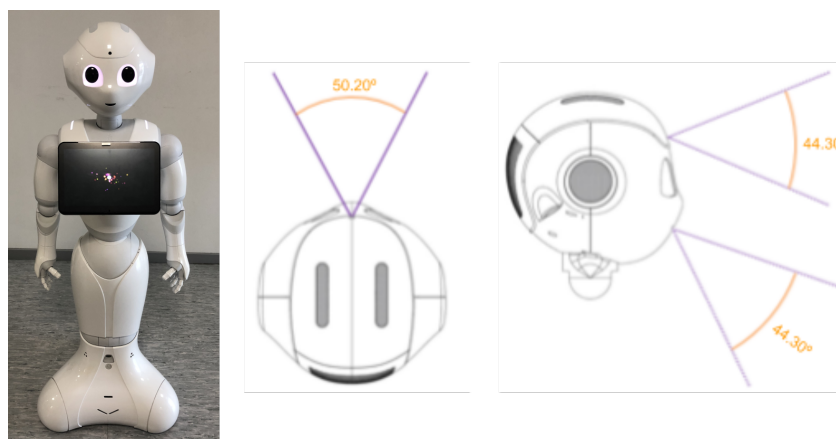
**Fig. 3.1:** Example of each class of the chosen subset of HMDB51 data.

**Tab. 3.1:** Characteristics of the subset of the HMDB51 dataset used in the approach.

Category	Videos	Width	Height	FPS
Brush hair	107			
Fencing	116			
Walk	282	Variable	240	30
Punch	126			
Smoke	109			
Cartwheel	103			

to engage with humans in socially effective ways. Using visual perception for human activity recognition will aid the robot to provide more accurate responses, enhancing its social capabilities. By identifying the action the user does, the robot will be able to tell when the user wishes to interact with it; for that purpose, a new database has been created.

The videos in the database have been recorded using the combined image obtained from Pepper<sup>1</sup> robot’s forehead cameras (see Fig. 3.2). The robot adjusts the orientation of its head according to the location of the face of the person appearing in its field of view.



**Fig. 3.2:** Pepper robot and the fields of view of its cameras.

As it can be seen in Tab. 3.2, it consists of six action categories, performed by 46 different people.

<sup>1</sup><https://www.softbankrobotics.com/emea/en/pepper>

**Tab. 3.2:** Characteristics of the dataset created with videos recorded by the robot.

Category	Description	Videos	Resolution	FPS
Come	Gesture to make the robot come to you	46	320 × 480	10
Five	Gesture of "high five"	45		
Handshake	Gesture of shaking hands with the robot	45		
Hello	Gesture to say hello to the robot	44		
Ignore	Ignore the robot, pass by	46		
Look at	Stare at the robot in front of it	46		

In Fig. 3.3 it can be seen a frame belonging to an example video for each of the classes of the database recorded with the robot. The robot is equipped with two RGB cameras located on its chin and forehead, and a combination of the two is used to obtain a larger range of vision in the recorded videos, as it can be appreciated in the examples.



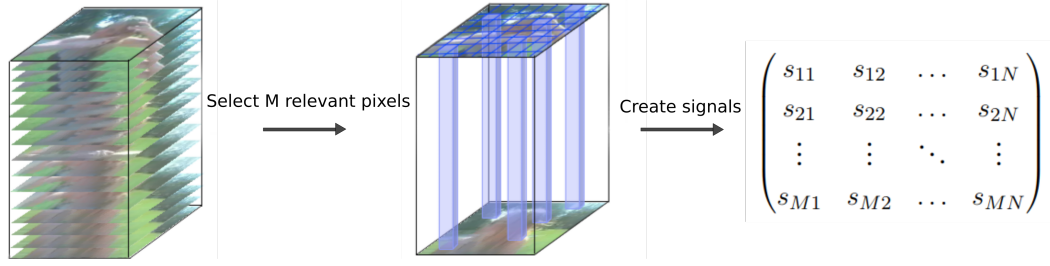
**Fig. 3.3:** Example of each class of the dataset recorded by Pepper.

### 3.3.2 Preprocessing

In order to apply the CSP algorithm we must obtain a list of instances, where each instance is a matrix of signals representing a video. Therefore, a video must be summarized into a matrix composed of signals and the data corresponding to each class is a list of these matrices.

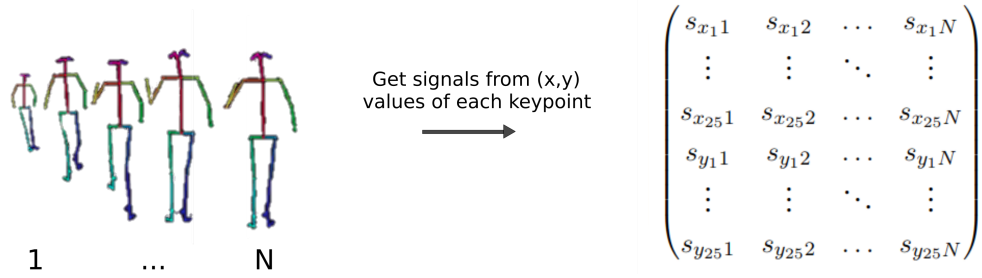
On the one hand, we have considered the value of a pixel of the image through time as a signal, where the value  $s_{11}$  is the value of pixel 1 in frame 1 and the value  $s_{1N}$  is the value of pixel 1 in the last frame ( $N$  indicates the number of frames of the video). This way we can create a matrix  $S \in \mathbb{R}^{M \times N}$  composed by  $M$  signals of length  $N$ , where  $M$  is the total number of pixels of a single image, and  $N$  is the time lapsed. To reduce the size of the matrix, a pixel selection is performed. Trying to choose the pixels that provide more information, those that have greater variance over time are selected, assuming that the pixels that do not vary correspond to static elements of the video that do not provide valuable information for the action recognition, such

as the background. The process of creating the signals matrix is shown graphically in Fig. 3.4.



**Fig. 3.4:** Process of creating the signals matrix from a video instance. First, the  $M$  most relevant pixels are selected according to their variance. Then, a matrix is created where rows represent pixels and columns indicate frames.

On the other hand, we have decided to extract the skeleton of the person performing the action using OpenPose (see Section 2.2.2) and compose the signals with the value of each keypoint over time. With a total of 25 keypoints extracted, 50 signals are created as each keypoint is represented by its  $(x, y)$  coordinate values. As shown in Fig. 3.5 for each keypoint two signals are created, one for  $x$  values and another one for  $y$  values, with  $N$  (number of frames) values each.



**Fig. 3.5:** Process of creating the signals matrix from skeletons extracted from a video instance. The skeletons are composed of 25 keypoints, each with its  $(x, y)$  coordinates values. A total of 50 signals are obtained.

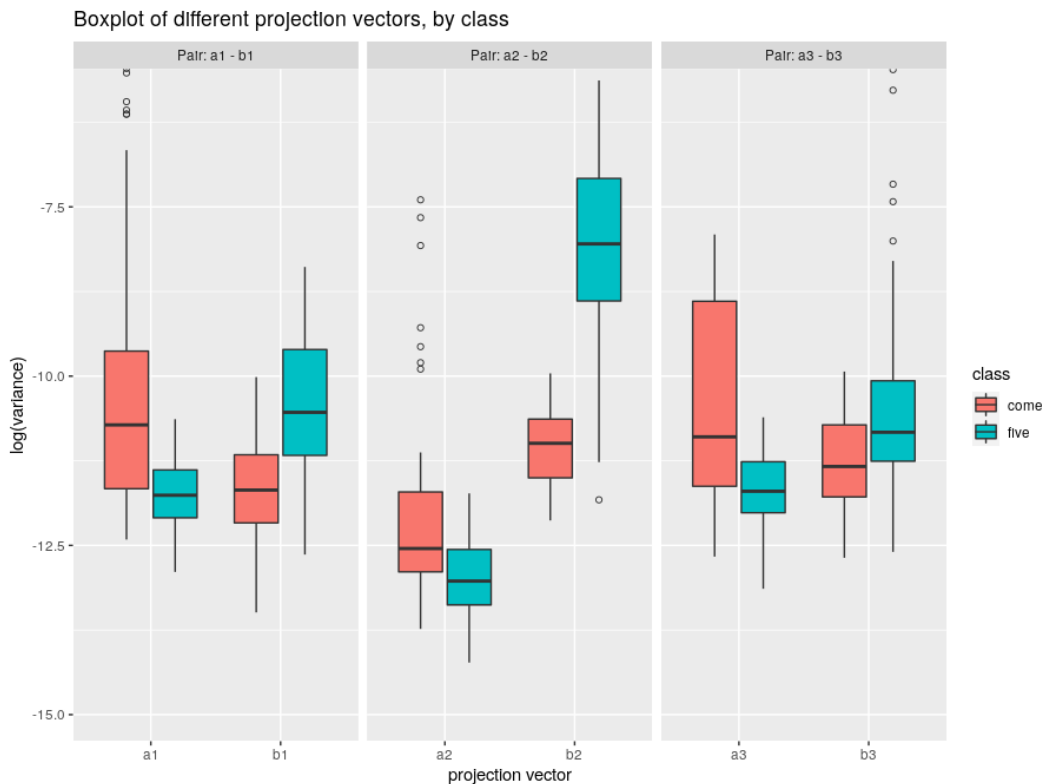
Let  $D_c$  be the data corresponding to class  $c$ . Equation 3.1 shows how the information corresponding to that class  $c$  should be represented, where  $L$  indicates the number of instances for that class,  $N$  the number of frames of the videos and  $M$  the number of signals.

$$D_C = \left[ \begin{array}{c} VIDEO_1 \\ \begin{pmatrix} s_{1,1,1} & s_{1,1,2} & \dots & s_{1,1,N} \\ s_{1,2,1} & s_{1,2,2} & \dots & s_{1,2,N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1,M,1} & \dots & \dots & s_{1,M,N} \end{pmatrix} \\ \dots, \\ VIDEO_L \\ \begin{pmatrix} s_{L,1,1} & s_{L,1,2} & \dots & s_{L,1,N} \\ s_{L,2,1} & s_{L,2,2} & \dots & s_{L,2,N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{L,M,1} & \dots & \dots & s_{L,M,N} \end{pmatrix} \end{array} \right] \quad (3.1)$$

### 3.3.3 Feature extraction

Once the  $D_c$  lists are obtained for every class  $c$ , the Common Spatial Patterns algorithm can be applied (see Section 2.1). For HMDB51 subset  $c \in \{\textit{brush hair}, \textit{cartwheel}, \textit{fencing}, \textit{punch}, \textit{smoke}, \textit{walk}\}$  while  $c \in \{\textit{come}, \textit{five}, \textit{handshake hello}, \textit{ignore}, \textit{look at}\}$  for the dataset recorded with Pepper.

The features used for classification are extracted from the preprocessed data after applying CSP and leaving only the selected  $2 \times q$  projection vectors. This projection takes into account the classes to which the data belongs and can only be applied to the instances of two classes at a time. For example, when working with the instances of classes  $c_1$  and  $c_2$ , after applying the algorithm, the selected  $2 \times q$  projection vectors are obtained, where the first  $q$  vectors ( $\mathbf{a} = 1 \dots q$  vectors) present high variance for instances of class  $c_1$  and low for those of class  $c_2$ , while the last  $q$  vectors ( $\mathbf{b} = (q + 1) \dots 2q$  vectors) yield the opposite. In the example of Fig. 3.6 an example for classes  $c_1 = \textit{come}$  and  $c_2 = \textit{five}$  with  $q = 3$  is displayed, where the differences between the variances of both classes after the projection can be seen.



**Fig. 3.6:** Log-variabilities of the projected signals on vectors  $\mathbf{a}_1$ ,  $\mathbf{a}_2$ ,  $\mathbf{a}_3$  and  $\mathbf{b}_1$ ,  $\mathbf{b}_2$ ,  $\mathbf{b}_3$ , separated by classes *come* and *five*. By construction, variabilities of the projections on vectors  $\mathbf{a}$  are big for units in class *come* and small for units in class *five*; the opposite pattern can be seen for projections on vectors  $\mathbf{b}$ .

Therefore, as the CSP projects signals to a space where the two classes are most separated by their variances, it is usual to use these variances to perform the classification. Even so, from the projected signals different features such as minimum, maximum or interquartile range (IQR) can be extracted. Thus, we have used both the variances and this extra information as features.

Besides, two matrix transformations have been applied using the signals projected by the CSP filter. The goal is to convert the data to a format that could be treated as an image and therefore be able to apply techniques of image classification. The two transformations are shown in equations 3.2,3.3, where  $A \in \mathbb{R}^{M \times N}$  is the matrix formed by the transformed video signals,  $M$  the number of signals and  $N$  the number of video frames. The first transformation corresponds to matrix multiplication and the second to the covariance matrix.

$$Q_1 = A * A^T \quad (3.2)$$

$$Q_2 = cov(A) = \frac{1}{n-1} \sum_{j=1}^n (A_j - \bar{A})(A_j - \bar{A})^T \quad (3.3)$$

The motivation behind these transformations is that one of the dimensions of the matrix is the number of signals ( $M$ ), but the other could be arbitrarily long, as it is the number of time steps or frames ( $N$ ). Applying a matrix multiplication by its transpose, the data is reduced to a manageable size. On the other side, the covariance matrix provides information about the global characteristics of the signals.

After applying these transformations, an  $M \times M$  matrix is obtained in each case, which is going to be treated as images and used for classification. Thus, through this approach, each video is summarized in a single image.

### 3.3.4 Classification

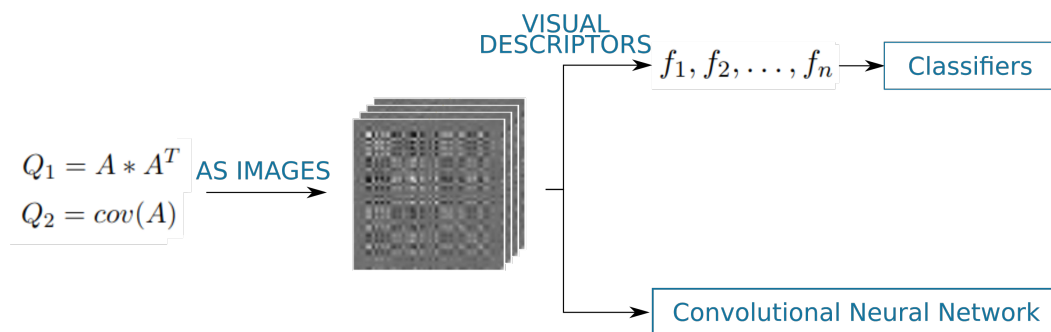
The classification method used depends on the features extracted from the signals obtained after applying the CSP algorithm to the original video signals.

**Features extracted from the projected signals.** In classical CSP applications, signal variances are used as features and classified by an LDA classifier. In our research, apart from extracting other characteristics such as the maximum or the IQR value, it has been decided to use different classifiers including Random Forest or KNN.

**Images obtained by applying matrix transformations to projected signals.** In the case of matrix transformation, since a summary image is obtained from each

video, image classification methods can be used for recognition. Specifically, it has been decided to use two different approaches (see Fig. 3.7):

- Convolutional Neural Networks (CNN). A CNN has been applied to classify the summary images obtained for each video. Its performance might drastically vary between several hyperparameter configurations, and therefore, in order to provide a fair comparison, we have used Keras Tuner Hypermodel, with a RandomSearch tuner to look for good configurations automatically. Convolutional layers, dropout layers, max pooling layers and a final dense layer of two units (as the classification is performed by pairs) make up the network. Adam [KB14] is used as optimizer and categorical cross-entropy as loss function. The learning rate, activation functions, number of filters and dropout rate hyperparameters have been tuned.
- Image descriptors with classical classifiers. Some commonly used visual descriptors have been applied to extract useful information from the created summary images, which are presented in Section 2.2.1. These descriptors describe visual features of images or videos, encoding interesting information into a list of numbers. Different classifiers have been trained with the feature vectors constructed from the descriptors, such as Bayesian Network (BN), J48 classification Tree, Naive Bayes (NB) or Support Vector Machine (SVM).



**Fig. 3.7:** Two different classification approaches based on the images obtained from matrices transformation.

**Comparison.** Methods commonly used for this task have been employed to make a comparison with our approaches. On the one hand, an LSTM network has been used as they are useful for classifications where temporality is a factor to be taken into account. On the other hand, the Histogram of Optical Flow method has been analysed for feature extraction, since this technique indicates the movement of the objects in the scene in consecutive frames.

### 3.3.5 Results

Regarding the dataset containing some of the classes from HMDB51 dataset, the best results are presented in Tab. 3.3, where the signals are created from most variable pixels of the videos. Although different classifiers have been tested, the best results correspond to Random Forest classifier for all the sets of features used. It has been decided to separate the results by the features used: (1) just the variances of the projected signals, (2) the variances along the maximum, minimum and interquartile range values from the projected signals, and (3) the histogram of optical flow features for comparison. These features have been used to train several classifiers but just the best results obtained for each set of features are displayed, which correspond to Random Forest classifier. When applying the CSP algorithm, different values of  $q$  variable have been used, achieving better accuracy values when  $q = 10$ .

- **var -  $q = 10$**  refers to the approach using  $q = 10$  vectors when applying CSP and the variances of the projected signals to perform the classification.
- **var, max, min, IQR -  $q = 10$**  represents the use of  $q = 10$  for CSP projection and the variances along with maximum, minimum and IQR values as features for classification.
- **Histograms of Optical Flow (HOF)**, the results obtained using HOF features for classification are also displayed to compare with the CSP-based approaches.

**Tab. 3.3:** Best results obtained for HMDB51 subset dataset, obtained by Random Forest classifier.

Categories	CSP-based approaches		
	(1) Variances	(2) More information	(3) Comparison
	var - $q = 10$	var, max, min, IQR - $q = 10$	HOF
BRUSH HAIR - CARTWHEEL	0.9445	0.9762	0.9535
BRUSH HAIR - FENCING	0.8651	0.9445	0.7576
BRUSH HAIR - PUNCH	0.8254	0.8254	0.8680
BRUSH HAIR - SMOKE	0.7302	0.7857	0.6894
BRUSH HAIR - WALK	0.7698	0.8174	0.8607
CARTWHEEL - FENCING	0.8889	0.8412	0.9380
CARTWHEEL - PUNCH	0.8492	0.9841	0.9504
CARTWHEEL - SMOKE	0.8730	0.9762	0.8992
CARTWHEEL - WALK	0.7937	0.9445	0.9103
FENCING - PUNCH	0.7391	0.9058	0.8889
FENCING - SMOKE	0.8030	0.8712	0.8106
FENCING - WALK	0.7391	0.9130	0.7553
PUNCH - SMOKE	0.7348	0.7424	0.8403
PUNCH - WALK	0.6000	0.6667	0.7912
SMOKE - WALK	0.5909	0.5758	0.7384
<b>MEAN</b>	0.7831	<b>0.8513</b>	0.8435

The outcomes depend on the classes being categorised because some pairs are easier to discern than others. For instance, the classes *brush hair* and *cartwheel* mostly obtain high accuracy values. Other classes, however, regardless of the technique or even the class to which they are compared, are harder to distinguish, such as *walk* for example. This could be due to the videos pertaining to the class, as they are taken from movies as mentioned above and these may be confusing or the information may not be well represented. The use of different characteristics does not significantly differ in terms of feature extraction methods.

Best results obtained for the action recognition dataset recorded with Pepper are displayed in Tab. 3.4. In this case, the signals are formed by the sequences of skeletons extracted using OpenPose. As a summary, it has been decided to show the results of the technique that obtains the highest accuracy score for each approach presented. In this case, the methods can be separated into three: (1) the classifiers trained by obtaining the features from the projected signals using the CSP, (2) the classifiers trained from the features extracted from the summary images obtained by the matrix transformation (both the visual descriptors and the convolutional network are included) and (3) the method used to compare the results of each of the approaches presented.

- **var+LDA** represents the best results achieved by the classification method of features extracted from projected signals (variances, in this case) and classical classifier (LDA).
- **MM<sup>T</sup>+PHOG+BN** refers to the best results obtained by the method of obtaining summary images from videos, which corresponds to matrix multiplication, the use of PHOG visual descriptors' features and Bayesian Network as classifier.
- **LSTM**, the results obtained by the comparative LSTM are also displayed.

Comparing the methods, it can be seen that the image summarization approach obtains better results for all class pairs. Moreover, both proposed approaches outperform the results obtained using a conventional LSTM. Concerning the classes, it can be said that the class *ignore* is the easiest to distinguish as its accuracy rate is high with every approach. On the other hand, the one that creates major confusion is the class *hello*, which is easily confused with other classes in the database.



**Tab. 3.4:** Best accuracy values obtained with the proposed methods, where the results of the best method of each type of approach are presented.

Pair of Categories	CSP-based approaches		
	(1) Features from projected signals	(2) Summary images from videos	(3) Comparison
	var + LDA	MM <sup>T</sup> + PHOG + BN	LSTM
COME - FIVE	0.7579	0.9780	0.8628
COME - HANDSHAKE	0.8668	0.9890	0.7739
COME - HELLO	0.5334	0.9444	0.7336
COME - IGNORE	0.9779	0.9891	0.9575
COME - LOOK AT	0.8678	1.0000	0.7849
FIVE - HANDSHAKE	0.9557	0.9889	0.8125
FIVE - HELLO	0.8208	0.9551	0.9125
FIVE - IGNORE	0.9668	1.0000	0.9789
FIVE - LOOK AT	0.9667	0.9780	0.8889
HANDSHAKE - HELLO	0.7431	0.9775	0.7108
HANDSHAKE - IGNORE	0.9889	1.0000	0.9764
HANDSHAKE - LOOK AT	0.8235	0.9560	0.8350
HELLO - IGNORE	0.9333	1.0000	0.9789
HELLO - LOOK AT	0.8445	1.0000	0.5733
IGNORE - LOOK AT	0.9889	0.9891	0.9775
MEAN	0.8691	<b>0.9830</b>	0.8505

### 3.4 Conclusions and future work

In this section, the work developed in the area of "Action Recognition" has been presented, which is collected in the following publications:

- **Video activity recognition: State-of-the-art.** Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Basilio Sierra, Igor Rodriguez, and Ekaitz Jauregi. In: *Sensors* 19.14 (2019), p. 3160.
- **Using Common Spatial Patterns to Select Relevant Pixels for Video Activity Recognition.** Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Basilio Sierra, Itziar Irigoien, Igor Rodriguez-Rodriguez, and Izaro Goienetxea. In: *Applied Sciences* 10.22 (2020), p. 8075.
- **Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns.** Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, Igor Rodríguez-Rodríguez, and Basilio Sierra. In: *Sensors* 20.8 (2020), p. 2436.
- **A New Approach for Video Action Recognition: CSP-Based Filtering for Video to Image Transformation.** Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, Igor Rodriguez, and Basilio Sierra. In: *IEEE Access* 9 (2021), pp. 139946–139957.

The approaches presented focus on the use of the Common Spatial Patterns algorithm for feature extraction in the task of identifying the different actions that actors are

representing. The algorithm tries to find an optimal filter that separates the instances corresponding to two different classes by their variances.

After projecting the signals corresponding to two classes using the CSP algorithm, it has been decided to extract different features from these signals. The standard use of the algorithm is to extract the variances and perform the classification using the LDA classifier. In addition to that, other features (min,max,iqr) have been extracted and several classifiers have been used to check if the recognition improves. To measure the suitability of CSP as a feature extraction method, other features have been extracted using the HOF method and the results obtained using both sets of features have been compared.

The key benefit of the suggested approach is that classical classifiers are very simple when compared to deep learning networks and there is no requirement for hyperparameter tuning. Since the models presented solely use the variances of the signals mainly, the feature set is also minimal.

Another pipeline for the classification after applying the CSP is also presented. Instead of extracting the features of the signals, a matrix transformation is performed with the matrices of these projected signals, which are then treated as images. Therefore, as each video is summarized into a single image, we are faced with an image classification problem. It has been decided to use both visual descriptors and a Convolutional network to perform this classification.

A comparison with Long Short-Term Memory neural network is conducted to validate the proposed CPS-based strategy, which yields better results. In conclusion, it is demonstrated that our objectives can be accomplished using a straightforward strategy that is typically employed for other tasks.

It is intended to extend the set of human activities in future work and incorporate the suggested approach into the real robot. As a result, the robot would be able to respond to a variety of actions performed in front of it in real-time, which could be interesting, for example, in social robotics.

# Sign Language Recognition

## 4.1 Introduction

A total of 1.5 billion individuals around the world have some degree of hearing loss, and of those, 430 million need rehabilitation services. Up to five out of every 1,000 babies are estimated to be born with hearing impairments or develop deafness shortly after birth, which can have a substantial influence on a child's growth and academic performance. Furthermore, hearing loss is a disabling condition that affects almost one out of every three adults over the age of 65. By 2050, this number is expected to increase to almost 700 million according to the World Health Organization (WHO), since due to the improper use of personal audio devices and exposure to loud noises at places like nightclubs, concerts or sporting events, millions of teenagers and young people are at danger of developing hearing loss.

The degree of hearing loss will result in some people requiring sign languages to communicate. According to the United Nations, about 72 million hard of hearing people use a sign language to communicate. In the European Union, France is the country with the highest number of signers with 300,000, followed by Spain with 100,000 and England with 77,000.

Sign Languages are a system of communication that involve the use of gestures, facial expression and body language to communicate. Although a commonly held belief by the general public is that there is a global and unique sign language, the reality is that there are over 300 different sign languages around the world, each language having its own lexicon, grammar and syntax. In Spain, since 2007, there are two sign languages officially recognised: Spanish and Catalan.

Despite the existence of sign languages, hard of hearing people often feel excluded as knowledge of sign languages is not widespread among hearing people, leading to frustration, isolation and loneliness. Whereas many people learn non-native spoken languages, sign languages are generally used by hard of hearing people, family members, professionals or people who have problems communicating with spoken languages. Therefore, the daily life interaction of people with hearing loss becomes more complicated without the assistance of an interpreter.

As sign languages are a key accessibility element for all these people, there have been recent attempts to develop different approaches in the area of automatic sign language recognition, ranging from wearable devices to computer vision or the study of sign language linguistics. Like spoken languages, sign languages have their own linguistic structures and can be challenging to translate for a variety of reasons. There are thousands of different signs in each sign language, many of which differ very slightly. In addition to the hand gestures used to convey a message, other elements such as hand placement, body alignment, and facial expression are also crucial. Therefore, an acceptable Sign Language Recognition system should take all these qualities into account.

## 4.2 State-of-the-art

In recent years, many different approaches have been developed to address the problem of automatic sign language recognition [RKE21; WK21]. Whereas American Sign Language (ASL) has been studied the most, approaches for recognising different sign languages have been presented, including the recognition of Spanish Sign Language [PMH17; VE+21; MMME21].

Sign languages can have complex grammatical structures, and a sign language recognition (SLR) system must involve both sign language linguistics and gesture recognition. Therefore, a SLR system must combine pattern matching, computer vision, natural language processing and linguistics in order to recognise the signs that are being performed and give a correct translation. In addition, although in some cases it is sufficient to use the information provided by the hands to recognise the sign being performed, facial features are considered essential, and in most cases these features are decisive in distinguishing similar signs even though few approaches use this information [VAKK08; KRD18].

Two main phases can be identified in this task: feature extraction and classification.

Over the years, different ways of extracting features have been followed. The extracted features can refer to the whole sign or to sub-units such as hand shape, hand placement, etc. The methods for extracting the features of the hands can be divided in intrusive, where there is a need to interfere with the signer to perform the feature extraction, and non-intrusive categories. Several examples of intrusive systems, for example with the use of colored or electronic gloves, have been developed [RM+18; KS18]. In terms of non-intrusive systems, different approaches have been implemented to obtain information related to the signer's body positions, for example using depth sensors, WiFi or pose extraction software [Ma+18; Pig+14; Par+20].

In the classification phase, Hidden Markov Models and Neural Networks are widely used. For instance, the authors of [Kum+17] use Coupled HMM based fusion of the data captured from two sensors (Leap Motion and Kinect) to perform the recognition. In [RKK21] the trajectories of hand motion are extracted after detecting the skin color from video frames and HMMs are used to perform the recognition of ASL signs. Researchers have made use of deep learning to achieve significant advances in sign language recognition, due to the effectiveness of deep learning techniques in a wide range of domains. Convolutional Neural Networks and Long Short-Term Memory networks are commonly used to learn spatial and temporal features, respectively [Mas+18; RKE20]. Lately, end-to-end solutions for sign language translation are being developed using Transformers [Cam+20].

Although advances in depth cameras, sensors and classification methods such as Deep Neural Networks are making the task of recognising sign languages more feasible, there is still much work to be done to overcome the complexity of this task. Moreover, while the performance of deep learning systems is impressive, they often lack the ability to explain their outputs in human-understandable terms and very few studies provide feedback to users [Pau+19].

## 4.3 Contributions

Within action recognition, more specific work has also been done on automatic sign language recognition. Currently, and despite the efforts made for the integration of people with different needs, hard of hearing people often have difficulties, either to use applications that have been designed for hearing people, or simply to communicate with their environment. Therefore, in terms of automatic sign language recognition, being able to recognise a representative subset of a sign language using a standard camera would allow the application of this technology to multiple devices. The research in this section has two main goals: (1) to endow a social robot with the capability of recognising the basic signs of the Spanish sign language and therefore be able to interact with the user using this skill, and (2) to add interpretability to the sign recognition task, which could be useful in several contexts, in special in the development of tutor apps.

The work developed on sign language recognition can be separated into two main ideas, which are explained below: CSP-based approach and hierarchical approach.

### 4.3.1 CSP-based approach

As in the action recognition task, the CSP algorithm has been used as a feature extraction method for sign language recognition. However, the relevant information for each task is different, since in sign language the hands play the most important role.

## Dataset

There are not a considerable number of sign language databases, and although there are some with a significant number of classes, most of them are not very large. In this instance, an Argentinian Sign Language (LSA) dataset composed of 64 signs known as the LSA64 dataset [Ron+16] is used.

The database contains a total of 3,200 videos, where 10 non-expert subjects perform each sign five times. Among the 64 classes, 22 signs are two-handed and 42 are single-handed. For our approach, it has been decided to use just the signs performed only with one hand. In Tab. 4.1 the classes to be recognised are listed along with the characteristics of the videos.

**Tab. 4.1:** Characteristics of the LAS64 dataset: one-handed signs.

Categories						Videos	Resolution	FPS
Opaque	Red2	Born	Water	Birthday	Deaf	2100 (50 per sign)	1920 × 1080	60
Red	Women	Learn	Food	Hungry	Candy			
Green	Enemy	Call	Argentina	Ship	Chewing-gum			
Yellow	Son	Skimmer	Uruguay	None	Shut down			
Bright	Man	Bitter	Country	Name	Buy			
Light-blue	Away	Sweet-milk	Last name	Patience	Realize			
Colors	Drawer	Milk	Where	Perfume	Find			

The subjects were recorded in both an indoor and outdoor environment while wearing black clothing and red and green gloves, which could be useful for hand segmentation although this is not carried out in the presented pipeline. It must also be mentioned that the subjects just concentrate on the movements of their hands when making the signs; they do not change their facial expression according to the sign they are performing. An example of a video from the database can be seen in Fig. 4.1.

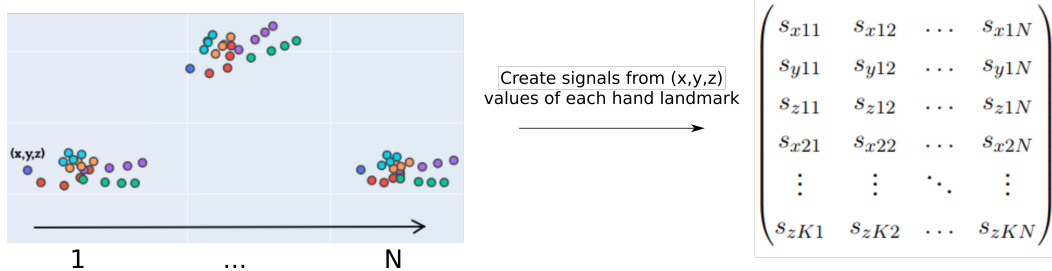


**Fig. 4.1:** Example of a frame sequence of a video from the LSA64 database.

## Presented approach

As previously mentioned, in the videos used, the gestures are performed considering the hands exclusively. Therefore, MediaPipe software has been used to extract the key points of the hand (see Section 2.2.2). Specifically, MediaPipe Hand Tracking solution has been used which offers 21 hand landmarks for each hand, each one composed of three values  $(x, y, z)$  where  $z$  indicates the depth in relation to the wrist.

In order to apply the CSP algorithm we need to convert each instance into a signal matrix. In this case, with each hand landmark value over time a signal of length  $N$  is formed, where  $N$  refers to the number of frames of the video. As each point is composed of three values, the three coordinates, with each one of them a signal is created. Therefore, a group of  $S \in \mathbb{R}^{3K \times N}$  signals are obtained composed of  $3 \times K$  signals of length  $N$ , being  $K$  the number of joints ( $K = 21$ ). This process is graphically displayed in Fig. 4.2.



**Fig. 4.2:** Process of creating the signals matrix from hand landmarks extracted from a video instance. The hand is composed of a total of  $K = 21$  landmarks, each one with its  $(x, y, z)$  coordinates values.

Once the set of signals are formed, the CSP algorithm is applied for each pair of classes in the same way as for the action recognition task, testing with different values for the  $q$  parameter.

In addition to the variances, the minimum, maximum and IQR values are added as features to be extracted from the projected signals. It is also worth mentioning that tests have been carried out taking into account the three coordinates  $(x, y, z)$  as well as only taking the  $(x, y)$  values, and that the videos have been transformed to black and white to check if this way the behavior of MediaPipe improves, getting a more reliable hand landmarks information.

Finally, different classifiers have been used for classification in order to make a comparison between them. The different parameters used during the pipeline are shown in the Tab. 4.2.

**Tab. 4.2:** Configuration of the parameters used in the classification process.

<b>Color space</b>	RGB — Black/white
<b>Used coordinates</b>	$(x, y, z)$ — $(x, y)$
<b>q value</b>	10 — 15
<b>Extracted features</b>	variances, maximum, minimum, IQR
<b>Classifiers</b>	BAGG — BN — J48 — KNN — NB — RF — SVM — MLP

## Results

The Tab. 4.3 shows a summary of the results obtained, where the average accuracy scores for each parameter value obtained with the different classifiers for all pairs of classes are shown, along with the median and the standard deviation.

**Tab. 4.3:** Obtained results for LSA64 dataset with different parameter values.

	<i>Color space</i>		<i>Used coordinates</i>		<i>q variable for CSP</i>	
	<b>RGB</b>	<b>B/W</b>	$(x, y, z)$	$(x, y)$	<b>q = 10</b>	<b>q = 15</b>
<b>Mean</b>	0.9139	<b>0.9546</b>	0.9288	<b>0.9397</b>	0.9250	<b>0.9436</b>
<b>Median</b>	0.9237	0.9691	0.9442	0.9468	0.9286	0.9487
$\sigma$	0.0374	0.0405	0.0431	0.0442	0.0418	0.0442

Since the signers are wearing colored gloves, it has been observed that MediaPipe occasionally isn't very precise. In an effort to boost its functionality, the original videos have been converted to black and white, which seems to work according to the results. There is no discernible difference between the coordinates selected for the classification but, as fewer features are used when simply considering  $(x, y)$  coordinates, this strategy is preferable. When applying the CSP technique, the  $q$  parameter controls how many feature vectors are used in the projection, better outcomes are attained with  $q = 15$ .

### 4.3.2 Hierarchical approach

A sign is defined by five elements (see Fig. 4.3), which are equivalent to the phonemes of oral languages, and together they compose the articulation of the sign:

- **Location:** the location where signs are performed, including the part of the body, the plane and the contact point.
- **Configuration:** the shape of the hand when performing a sign.



- Orientation: orientation of the hands involved on the articulation of the sign with respect to the body of the signer.
- Movement: the movement usually involved when performing a sign.
- Non-manual component: body position and facial expression.



**Fig. 4.3:** Elements which compose the grammar of sign language.

Taking these elements into account, the recognition of signs is based on the definition of these signs, more specifically on the changes in the shape of the hand (configurations) when performing the signs. A hierarchical approach for Spanish Sign Language (LSE, for its acronym in Spanish) recognition is presented, using the definitions extracted from [GS+16], where the signs are decomposed into hand configurations to perform the classification.

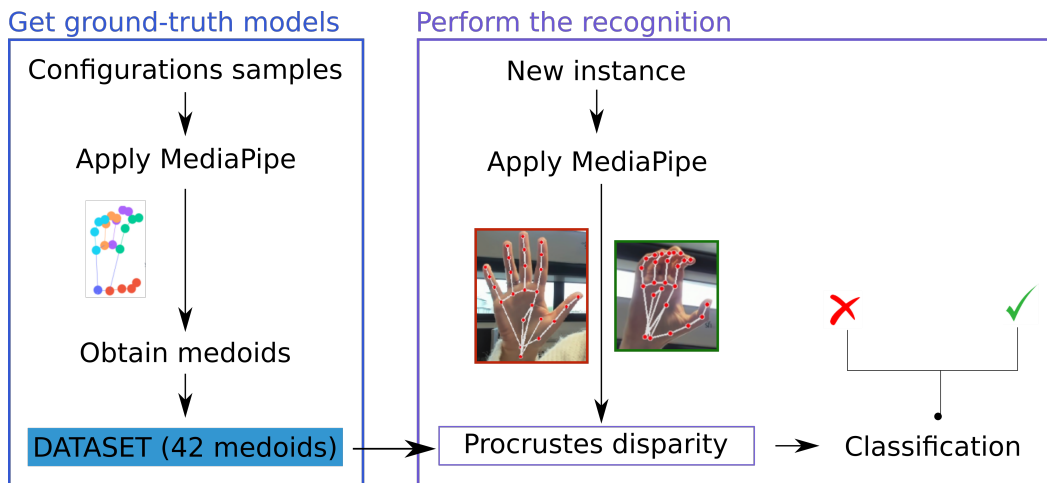
The ultimate goal of this approach is to develop a tutor for learning sign language, capable of giving feedback to users on their performance.

## Recognising configurations

In the Spanish Sign Language there are 42 different configurations [FM+19] and the goal is to develop a tutor to help the user to learn all of them in order to be able to produce and recognise different signs when communicating in sign language. Automatic recognition is challenging due to the high degree of visual similarity among some subsets of configurations.

The key idea is to use Procrustes analysis [Gow75] to compare the hand landmarks extracted from the user performing the configuration to previously saved models to give a prediction. As it can be seen in Fig. 4.4, the followed recognition process can be separated into two parts.

**Get ground-truth models.** First, the dataset of the ground-truth models must be created composed of a total of 42 models (one per configuration). For that purpose, a person with basic formal education in Spanish Sign Language performed 50 times each configuration in front of a Logitech BRIO 4K Ultra HD Webcam and in good lighting conditions. From these frames, the hand landmarks are extracted using



**Fig. 4.4:** Process of recognition of configurations. First, models of each configuration are obtained which are then used to compare to the new case to give a prediction.

MediaPipe. Then, the medoid of the repetitions with respect to the Procrustes disparity is computed and saved as ground-truth. This is only computed once, in order to create the medoids dataset before performing the classification.

**Perform the recognition.** After obtaining the hand landmarks of the new instance applying MediaPipe, this instance is compared to the previously saved models. This comparison is made using the Procrustes disparity, which consists of performing a series of transformations (scaling, rotating, mirroring) to minimise the difference between two shapes. This way, the Procrustes similarities between the configuration that is being performed and the 42 saved models are calculated and the configuration which achieves the lowest difference is predicted, that is, the closest configuration is used to decide the label of the performed configuration.

## Web Application

As part of the goal of developing a Spanish Sign Language tutor, a web application has been built to help users learn the 42 configurations of the LSE, which is available as open-source on GitHub: [https://github.com/rsait/LSE\\_tutor](https://github.com/rsait/LSE_tutor). The functionalities offered on the web are explained next.

1. **Learn configurations.** The user is able to choose the configuration to practice, along with the hand with which it is going to be performed. An image of the chosen hand shape and a 3D graph with the hand landmarks of the pre-saved model for that configuration are shown in order to help the user. Once the user starts performing the configuration, a green or red background is set to let the user know if it is being carried out correctly or not, respectively.

2. **Analyse the performance.** The user can analyse the errors he/she has made thanks to the record of the mistakes that it is saved. This way, the user can focus on the configurations he/she misses the most.
3. **Explore signs.** To motivate the learning of configurations a set of 196 signs have been added, and when the user select one of them the configurations corresponding to it are displayed in the same order in which they are carried out in the sign. The sequence of configurations must be performed correctly as if the sign was actually being executed. When the configuration is correct, the background turns green and the red background moves to the next one.

The appearance of the explained functionalities are shown in Fig. 4.5.

## Recognising signs

As the final goal is to recognise the signs of the Spanish Sign Language (LSE, for its acronym in Spanish), a limited group of signs has been selected in order to test whether the decomposition of these signs into configurations is beneficial for recognition. The idea of decomposing the signs into their constituents, as mentioned before, was born mainly with the purpose of being able to explain to the users the reasons behind the obtained results, thus being able to give feedback on their performance.

Therefore, the five signs from the LSE indicated in Tab. 4.4 have been selected, which are composed of eight different configurations. The recognition process is slightly different from the one explained above, and to implement it, two different databases have been created: (1) with images corresponding to the configurations and (2) with videos of the chosen signs. Six people participated in the creation of the configurations database, while the signs database was created with five people and each video has 25 frames.

As a first step, to recognise the configurations a set of classical classifiers (KNN, RF, SVM) have been trained with the hand landmarks extracted by MediaPipe. In addition to these keypoints, it has been decided to include some distances between certain fingers (between fingertips and from fingertips to thumbtip) as features since the distances are independent of the spatial location of the hand and, therefore, expected to be useful when performing configuration classification.


Two different approaches are proposed for the classification of the signs as shown in Fig. 4.6, where the followed pipeline is described.

### Choose which hand you are using

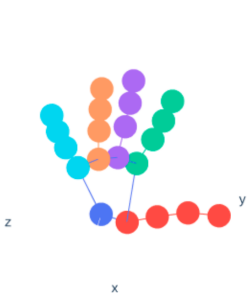
Right hand  Left hand

### Choose a configuration to practice

1



- Hand
- Thumb
- Index
- Middle
- Ring
- Pinky



1



## PERFORMANCE.

**The 57.69% of the trials were well performed.**

These are the ten most confusing configurations in your attempts.

	Chosen configuration	Performed configuration	% Wrong performances
<input type="radio"/>	1	3	23.08
<input checked="" type="radio"/>	1	2	19.23

**You have selected to practice configuration 1    But you have performed configuration 2**

[Continue learning](#)

2


### Choose which hand you are using


Right hand  Left hand


**Topic:**

**Word:**

These are the 3 configurations you must perform:















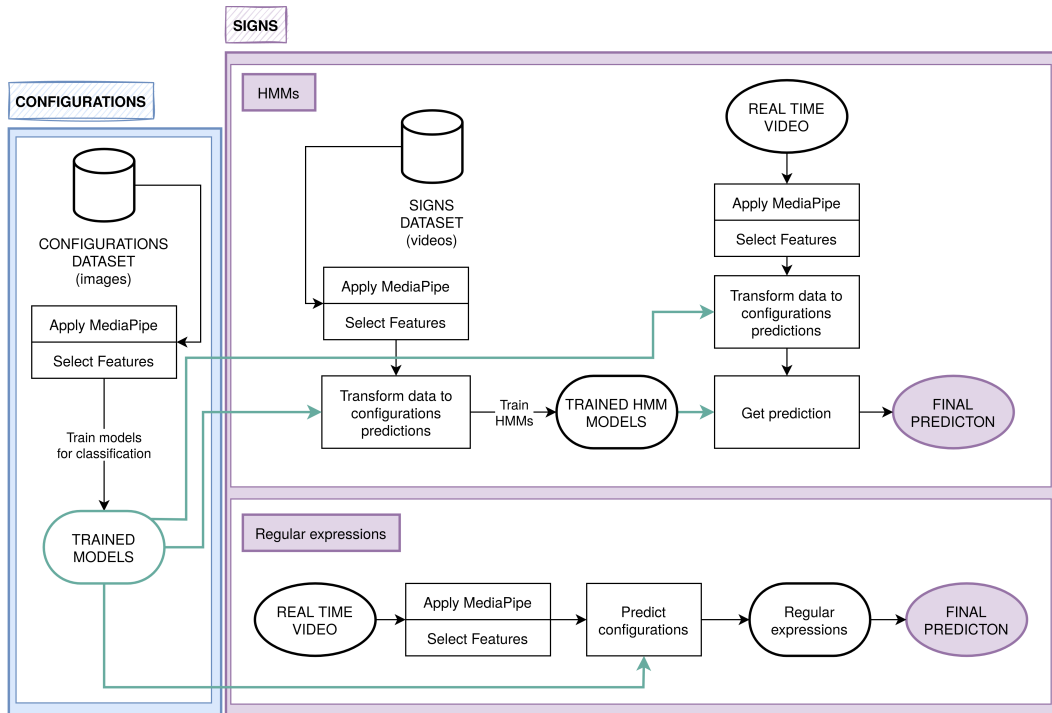


3

**Fig. 4.5:** The different functionalities of the web application. (1) Learn configurations. (2) Check the performance. (3) Search signs composed of different configurations.

**Tab. 4.4:** Definition of the selected signs and number of instances used to create the databases.

SIGN	NUMBER OF VIDEOS	INITIAL HAND CONFIGURATION	NUMBER OF IMAGES	FINAL HAND CONFIGURATION	NUMBER OF IMAGES
Well ( <i>Bien</i> )	175	 10	961	 1	1019
Happy ( <i>Contento</i> )	176	 20	875	 18	900
Man ( <i>Hombre</i> )	174	 21	915	 21	915
Woman ( <i>Mujer</i> )	175	 13	938	 14	958
Listener ( <i>Oyente</i> )	175	 3	991	 3	991
<b>TOTAL VIDEOS FOR SIGNS DATASET</b>		<b>875</b>	<b>TOTAL IMAGES FOR CONFIGURATIONS DATASET</b>		<b>9463</b>



**Fig. 4.6:** Followed pipeline for sign recognition. On the left, the classification process for configurations is shown. On the right, the two different approaches for sign recognition are displayed, where HMMs or regular expressions are used.

1. **Regular expressions.** Once the configurations classifier has been trained, it can be used to classify each frame of the video into a configuration, summarizing the video in an array of configurations. Based on the definitions of the signs, five regular expressions have been defined (one per sign) with which the vector of configurations is evaluated to decide which sign has been performed. In Tab. 4.5 the regular expression corresponding to each of the selected signs are displayed, which match with the definition of the signs.

**Tab. 4.5:** Regular expressions corresponding to each of the selected signs.

Sign	Regular expression
Well (bien)	(10) + (1)+
Happy (contento)	(20) + (18)+
Man (hombre)	(21) + (21)+
Woman (mujer)	(13) + (14)+
Listener (oyente)	(3) + (3)+

2. **Hidden Markov Models (HMMs).** On the other hand, a set of HMMs are trained to perform the sign recognition. To do so, each video is first transformed into a probability matrix obtained with the predictions of the configurations model. Specifically, each video is converted into the matrix  $V \in \mathbb{R}^{25 \times 8}$  shown in Eq. 4.1, where  $P_{i,j}$  indicates the probability of frame  $i$  to correspond to configuration  $j$ .

$$V = \begin{pmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,8} \\ P_{2,1} & P_{1,2} & \dots & P_{2,8} \\ \vdots & \vdots & \ddots & \vdots \\ P_{25,1} & \dots & \dots & P_{25,8} \end{pmatrix} \quad (4.1)$$

After transforming the videos, an HMM is trained with the instances corresponding to each class, thus obtaining five different HMMs. Based on the definition of the signs these HMMs are defined with two states, corresponding to the initial and the final configuration, respectively. When classifying a new video, it is first transformed into the probability matrix using the configurations model, which is then used as input for all the trained HMMs. The class to be predicted is determined by the HMM with the best score.

## 4.4 Conclusions and future work

In this section, the work developed in the area of "Sign Language Recognition" has been presented, which is collected in the following publications:

- **Sign Language Recognition by Means of Common Spatial Patterns.** Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, and Basilio Sierra. In: *2021 The 5th International Conference on Machine Learning and Soft Computing*. 2021, pp. 96–102.
- **Sign Language Recognition by Means of Common Spatial Patterns: An Analysis.** Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, and Basilio Sierra. In: *Plos one* 17.10 (2022), e0276941.
- **Towards an Interpretable Spanish Sign Language Recognizer.** Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, and Basilio Sierra. In: *ICPRAM*. 2022, pp. 622–629.
- **A Hierarchical Approach for Spanish Sign Language Recognition: From Weak Classification to Robust Recognition System.** Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, and Basilio Sierra. In: *Proceedings of SAI Intelligent Systems Conference*. Springer. 2022, pp. 37–53.
- **HAKA: HierArchical Knowledge Acquisition in a Sign Language Tutor.** Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, and Basilio Sierra. In: *Expert Systems with Applications* 215 (2022), p. 119365.

On the one hand, it has been shown that the previously presented method used for action recognition is also useful for sign language recognition. In this case, the CSP algorithm is applied with the signals formed with the hand landmarks in order to separate the instances belonging to two different signs of the Argentinian sign language based on their variances.

Comparing our approach to deep learning approaches, where hyperparameters need to be fine-tuned which requires running numerous training epochs with each set of candidate hyperparameter values, one advantage is that the CSP has a closed form and can therefore be computed without using iterative methods. The approach described uses just five hyperparameters (see Tab. 4.2), significantly less than a typical deep learning hyperparameter tuning task.

Several details could be considered for future work. The database used (LSA64) includes signs performed with both hands that have not been taken into account so far. It would be interesting to be able to classify two-handed signs. Apart from the hands, the facial expression plays a key role in recognising the signs. Therefore,

adding face information (for instance, using FaceMesh<sup>1</sup> from MediaPipe) would be necessary to build a strong sign language recognition system.

In addition, other techniques (deep learning, for example) could be applied on the signals projected by the CSP algorithm, instead of performing the classification on the features extracted from them.

Regarding the hierarchical approach, we present a bottom-up approach, where the signs are decomposed into constituents to perform the recognition. The aim is to obtain better interpretability of the decisions made by the recogniser, which is difficult to achieve with end-to-end deep learning approaches. The tutor is intended to provide explanations of the classification and thus help the user to improve his/her performance.

To provide the tutor with an explainability module offers the option of reasoning the generated answers. Whether the sign is correctly executed or not, it will be possible to indicate the reason for this decision. This is especially beneficial when the user performance is incorrect, as the explanations provided will allow the user to pay specific attention to the mistakes he/she is making and correct the aspects that make the sign an incorrect replica of the real sign. Therefore, future work should focus on improving the explainability of the system, as this is crucial when trying to help people who are learning a sign language.

When it comes to the recognition of configurations through Procrustes similarity, adding a new configuration or modifying already defined configurations is straightforward. The new model simply needs to be added to the medoids database, thereby the process can be considered a "lazy" method where no retraining is required when adding new information.

A web application has also been developed to be used as a tutor to learn the 42 configurations of Spanish sign language. This application can be found on GitHub ([https://github.com/rsait/LSE\\_tutor](https://github.com/rsait/LSE_tutor)) and a docker image is available so that the user can easily launch it.

The next step would be to recognise whole signs consisting of a succession of these 42 hand configurations. Apart from the hand landmarks, specific body keypoints and the distance between them, together with facial expression, would also need to be added as features. Alternatively, while this additional information is not considered, a sentence indicating the part of the body where the sign should be performed could be added for informational purposes.

---

<sup>1</sup>[https://google.github.io/mediapipe/solutions/face\\_mesh.html](https://google.github.io/mediapipe/solutions/face_mesh.html)



# Bibliography

- [Ari+19a] Sheeraz Arif, Jing Wang, Tehseen Ul Hassan, and Zesong Fei. “3D-CNN-Based Fused Feature Maps with LSTM Applied to Action Recognition”. In: *Future Internet* 11.2 (2019), p. 42.
- [Ari+19b] S Arivazhagan, R Newlin Shebiah, R Harini, and S Swetha. “Human action recognition from RGB-D data using complete local binary pattern”. In: *Cognitive Systems Research* 58 (2019), pp. 94–104.
- [Bed+20] Djamila Romaiissa Beddiar, Brahim Nini, Mohammad Sabokrou, and Abdenour Hadid. “Vision-based human activity recognition: a survey”. In: *Multimedia Tools and Applications* 79.41 (2020), pp. 30509–30555.
- [CA09] Chia-Chih Chen and JK Aggarwal. “Recognizing human action from a far field of view”. In: *2009 Workshop on Motion and Video Computing (WMVC)*. IEEE. 2009, pp. 1–7.
- [Cam+20] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. “Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10023–10033.
- [Cao+17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7291–7299.
- [Cha+09] Rizwan Chaudhry, Avinash Ravichandran, Gregory Hager, and René Vidal. “Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 1932–1939.
- [Cho+18] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. “PoTion: Pose MoTion Representation for Action Recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7024–7033.
- [DLL20] Cheng Dai, Xingang Liu, and Jinfeng Lai. “Human action recognition using two-stream attention based LSTM networks”. In: *Applied soft computing* 86 (2020), p. 105820.

- [FK70] Keinosuke Fukunaga and Warren LG Koontz. “Application of the Karhunen-Loève Expansion to Feature Selection and Ordering”. In: *IEEE Transactions on Computers* 100.4 (1970), pp. 311–318.
- [FM+19] Miriam Fernández Martín, María Isabel Moreno Ribera, Dámaris I. Caulín Bonilla, and Miguel Jiménez Santiago. *Lengua de signos*. Editorial Síntesis, 2019.
- [Gow75] John C Gower. “Generalized Procrustes Analysis”. In: *Psychometrika* 40.1 (1975), pp. 33–51.
- [GS+16] Eva Gutierrez-Sigut, Brendan Costello, Cristina Baus, and Manuel Carreiras. “LSE-Sign: A lexical database for Spanish Sign Language”. In: *Behavior Research Methods* 48.1 (2016), pp. 123–137.
- [HHP17] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. “Going Deeper into Action Recognition: A Survey”. In: *Image and vision computing* 60 (2017), pp. 4–21.
- [KB14] Diederik P Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [KF22] Yu Kong and Yun Fu. “Human Action Recognition and Prediction: A Survey”. In: *International Journal of Computer Vision* 130.5 (2022), pp. 1366–1401.
- [KRD18] Pradeep Kumar, Partha Pratim Roy, and Debi Prosad Dogra. “Independent Bayesian classifier combination based sign language recognition using facial expression”. In: *Information Sciences* 428 (2018), pp. 30–48.
- [KS18] Nayan M Kakoty and Manalee Dev Sharma. “Recognition of Sign Language Alphabets and Numbers based on Hand Kinematics using A Data Glove”. In: *Procedia Computer Science* 133 (2018), pp. 55–62.
- [Kue+11] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. “HMDB: A large video database for human motion recognition”. In: *2011 International conference on computer vision*. IEEE. 2011, pp. 2556–2563.
- [Kum+17] Pradeep Kumar, Himaanshu Gauba, Partha Pratim Roy, and Debi Prosad Dogra. “Coupled HMM-based multi-sensor data fusion for sign language recognition”. In: *Pattern Recognition Letters* 86 (2017), pp. 1–8.
- [LAC11] Kanokphan Lertniphonphan, Supavadee Aramvith, and Thanarat H Chalid-abhongse. “Human action recognition using direction histograms of optical flow”. In: *2011 11th International Symposium on Communications & Information Technologies (ISCIT)*. IEEE. 2011, pp. 574–579.
- [LGH19] Ji Lin, Chuang Gan, and Song Han. “TSM: Temporal Shift Module for Efficient Video Understanding”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7083–7093.
- [Lin+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. “Microsoft COCO: Common Objects in Context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

- [LLC17] Mengyuan Liu, Hong Liu, and Chen Chen. “Robust 3D Action Recognition through Sampling Local Appearances and Global Distributions”. In: *IEEE Transactions on Multimedia* 20.8 (2017), pp. 1932–1947.
- [Lug+19] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, et al. “MediaPipe: A Framework for Building Perception Pipelines”. In: *arXiv preprint arXiv:1906.08172* (2019).
- [Ma+18] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. “SignFi: Sign Language Recognition Using WiFi”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1 (2018), pp. 1–21.
- [Mas+18] Sarfaraz Masood, Adhyan Srivastava, Harish Chandra Thuwal, and Musheer Ahmad. “Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN”. In: *Intelligent Engineering Informatics*. Springer, 2018, pp. 623–632.
- [MMME21] Ester Martinez-Martin and Francisco Morillas-Espejo. “Deep Learning Techniques for Spanish Sign Language Interpretation”. In: *Computational Intelligence and Neuroscience 2021* (2021).
- [NFF07] Juan Carlos Niebles and Li Fei-Fei. “A Hierarchical Model of Shape and Appearance for Human Action Classification”. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.
- [NYV18] Saima Nazir, Muhammad Haroon Yousaf, and Sergio A Velastin. “Evaluating a bag-of-visual features approach using spatio-temporal features for action recognition”. In: *Computers & Electrical Engineering* 72 (2018), pp. 660–669.
- [Par+20] Maria Parelli, Katerina Papadimitriou, Gerasimos Potamianos, Georgios Pavlakos, and Petros Maragos. “Exploiting 3D Hand Pose Estimation in Deep Learning-Based Sign Language Recognition from RGB Videos”. In: *European Conference on Computer Vision*. Springer, 2020, pp. 249–263.
- [Pau+19] Prajwal Paudyal, Junghyo Lee, Azamat Kamzin, et al. “Learn2Sign: Explainable AI for Sign Language Learning.” In: *IUI Workshops*. 2019.
- [Pig+14] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen. “Sign Language Recognition Using Convolutional Neural Networks”. In: *European conference on computer vision*. Springer, 2014, pp. 572–578.
- [PMH17] Zuzanna Parcheta and Carlos-D Martínez-Hinarejos. “Sign Language Gesture Recognition Using HMM”. In: *Iberian conference on pattern recognition and image analysis*. Springer, 2017, pp. 419–426.
- [RKE20] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. “Hand sign language recognition using multi-view hand skeleton”. In: *Expert Systems with Applications* 150 (2020), p. 113336.
- [RKE21] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. “Sign Language Recognition: A Deep Survey”. In: *Expert Systems with Applications* 164 (2021), p. 113794.
- [RKK21] Partha Pratim Roy, Pradeep Kumar, and Byung-Gyu Kim. “An Efficient Sign Language Recognition (SLR) System Using Camshift Tracker and Hidden Markov Model (HMM)”. In: *SN Computer Science* 2.2 (2021), pp. 1–15.

- [RM+18] Paul D Rosero-Montalvo, Pamela Godoy-Trujillo, Edison Flores-Bosmediano, et al. “Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques”. In: *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*. IEEE. 2018, pp. 1–5.
- [RM+19] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Basilio Sierra, Igor Rodríguez, and Ekaitz Jauregi. “Video Activity Recognition: State-of-the-Art”. In: *Sensors* 19.14 (2019), p. 3160.
- [RM+20a] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, Igor Rodríguez-Rodríguez, and Basilio Sierra. “Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns”. In: *Sensors* 20.8 (2020), p. 2436.
- [RM+20b] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Basilio Sierra, et al. “Using Common Spatial Patterns to Select Relevant Pixels for Video Activity Recognition”. In: *Applied Sciences* 10.22 (2020), p. 8075.
- [RM+21a] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, Igor Rodríguez, and Basilio Sierra. “A New Approach for Video Action Recognition: CSP-Based Filtering for Video to Image Transformation”. In: *IEEE Access* 9 (2021), pp. 139946–139957.
- [RM+21b] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, and Basilio Sierra. “Sign Language Recognition by Means of Common Spatial Patterns”. In: *2021 The 5th International Conference on Machine Learning and Soft Computing*. 2021, pp. 96–102.
- [RM+22a] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, and Basilio Sierra. “Sign Language Recognition by Means of Common Spatial Patterns: An Analysis”. In: *Plos one* 17.10 (2022), e0276941.
- [RM+22b] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, and Basilio Sierra. “Towards an Interpretable Spanish Sign Language Recognizer”. In: *ICPRAM*. 2022, pp. 622–629.
- [RMMOS22a] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, and Basilio Sierra. “A Hierarchical Approach for Spanish Sign Language Recognition: From Weak Classification to Robust Recognition System”. In: *Proceedings of SAI Intelligent Systems Conference*. Springer. 2022, pp. 37–53.
- [RMMOS22b] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, and Basilio Sierra. “HAKA: Hierarchical Knowledge Acquisition in a Sign Language Tutor”. In: *Expert Systems with Applications* 215 (2022), p. 119365.
- [Rod+22] Itsaso Rodríguez, Itziar Irigoien, Basilio Sierra, and Concepción Arenas. “The R Journal: dbcsp: User-friendly R package for Distance-Based Common Spatial Patterns”. In: *The R Journal* 14.3 (2022). <https://doi.org/10.32614/RJ-2022-044>, pp. 80–94.
- [Ron+16] Franco Ronchetti, Facundo Quiroga, César Armando Estrebow, Laura Cristina Lanzarini, and Alejandro Rosete. “LSA64: An Argentinian Sign Language Dataset”. In: *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*. 2016.

- [Shi+19] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. “Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 12026–12035.
- [STC18] Sowndarya Satyamurthi, Jing Tian, and Matthew Chin Heng Chua. “Action recognition using multi-directional projected depth motion maps”. In: *Journal of Ambient Intelligence and Humanized Computing* (2018), pp. 1–7.
- [TS08] Du Tran and Alexander Sorokin. “Human Activity Recognition with Metric Learning”. In: *European conference on computer vision*. Springer. 2008, pp. 548–561.
- [Ull+17] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. “Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features”. In: *IEEE access* 6 (2017), pp. 1155–1166.
- [VAKK08] Ulrich Von Agris, Moritz Knorr, and Karl-Friedrich Kraiss. “The significance of facial features for automatic sign language recognition”. In: *2008 8th IEEE international conference on automatic face & gesture recognition*. IEEE. 2008, pp. 1–6.
- [VE+21] Manuel Vázquez-Enríquez, Jose L Alba-Castro, Laura Docío-Fernández, and Eduardo Rodríguez-Banga. “Isolated Sign Language Recognition with Multi-Scale Spatial-Temporal Graph Convolutional Networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3462–3471.
- [Wan+15] Pichao Wang, Wanqing Li, Zhimin Gao, et al. “Action Recognition from Depth Maps Using Deep Convolutional Neural Networks”. In: *IEEE Transactions on Human-Machine Systems* 46.4 (2015), pp. 498–509.
- [WK21] Ankita Wadhawan and Parteek Kumar. “Sign Language Recognition Systems: A Decade Systematic Literature Review”. In: *Archives of Computational Methods in Engineering* 28.3 (2021), pp. 785–813.
- [YLZ19] Guangle Yao, Tao Lei, and Jiandan Zhong. “A review of Convolutional-Neural-Network-based action recognition”. In: *Pattern Recognition Letters* 118 (2019), pp. 14–22.
- [Zha+20] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, et al. “MediaPipe Hands: On-device Real-time Hand Tracking”. In: *arXiv preprint arXiv:2006.10214* (2020).



# Part II

---

Conclusions





## Conclusions and future work

In this PhD dissertation the work developed in two tasks is presented: video action recognition and sign language recognition. Both are related to the same field of research, computer vision, and the goal in both tasks consists on developing a system for human-computer interaction.

The use of the Common Spatial Patterns algorithm as a previous step before classification is presented. Briefly, this algorithm tries to find an optimal spatial filter to separate instances belonging to two different classes, where the first  $q$  vectors of the filter  $\mathbf{W}$  produce large variances for class  $C_1$  and low variances for class  $C_2$ , while the last  $q$  vectors produce the opposite. The filter  $\mathbf{W}$  is obtained by computing the generalised eigendecomposition of the covariance matrices of both classes.

The classical CSP algorithm is based on the Euclidean distance and although this has been used in the presented approaches, an extension of the classical CSP has been developed and is available open-source in CRAN called **dbcsp**. This package called Distance-Based Common Spatial Patterns allows the use of any distance when applying the CSP to project the signals, which is obtained computing the average distance-based covariance matrices.

For action recognition, the information to feed the CSP algorithm is obtained in two different ways: the signals are obtained through the pixel values of the image in time or through the skeleton values of the user performing the action in time. Once the CSP algorithm is applied and the features are obtained as explained in Section 2.1, different classifiers are applied. Furthermore, the videos have been summarised into images by performing matrix transformations with the projected signals, thus allowing the application of image classification techniques. The presented approaches have been compared to commonly used techniques in video action recognition.

Compared to deep learning networks where the number of hyperparameters is very large, a simple approach is presented making use of classical classifiers and the CSP, where only the  $q$  parameter needs to be tuned. In addition, as the variances of the signals are used after applying the CSP, the feature set is small when training the classifiers. However, this method has two main limitations:

- Binary classification: the algorithm can only be applied to two classes at a time, but class binarization techniques, such as One-vs-One (OVO) or One-vs-All (OVA), can be applied for multi-class problems.
- Computational cost: as explained in Section 2.1, CSP algorithm applies matrix operations, computing the generalized eigendecomposition over the covariance matrices of two classes. Thus, the execution time increases with the size of the matrices. In order to avoid high computational cost, a low image resolution has been used when creating the signals from the image pixels.

As future work, two main paths are considered. On the one hand, the aim is to increase the number of classes to be recognised, applying binarization techniques for multi-class classification. On the other hand, we intend to implement the recognition system in the real robot, endowing it with the capability to react on the basis of the action performed by the user.

In respect of sign language recognition, a new approach is presented trying to address the issue of explainability when performing the classification. In short, a tutor is presented to help users who are learning Spanish Sign Language (LSE). To address this problem, the signs are decomposed into their constituents to perform the recognition, based on the definition of the signs. Specifically, hand configurations (the shape of the hand) and the movement from one configuration to another are used to classify the signs. The idea of this approach is to be able to indicate the reasons behind the obtained results, and thus improve both the classification process and the feedback users receive.

As a result, a web application has been developed and is available on GitHub ([https://github.com/rsait/LSE\\_tutor](https://github.com/rsait/LSE_tutor)), where users can learn the 42 configurations of the LSE, as well as analyse different signs composed of these configurations. Through this application users are able to check if they are performing the configurations correctly or the most common mistakes they make in real time.

It is intended to continue working on the LSE tutor in the future, providing it with greater recognition capacity in order to improve the explanations given to the user on how to perform the signs. In addition, as signs are defined by several elements and not only by the shape of the hand (see Fig. 4.3), new features should be added to classify the signs. Besides, the application currently does not have the functionality to recognise signs, which should be included when a more complete recognition system is achieved. Finally, this tutor could also be integrated into a real humanoid robot, being the robot the one that gives the feedback to the users.

# Part III

---

Publications



# dbcsp: User-friendly R package for Distance-Based Common Spatial Patterns

<b>Title:</b>	dbcsp: User-friendly R package for Distance-Based Common Spatial Patterns
<b>Authors:</b>	I. Rodríguez, I. Irigoien, B. Sierra, C. Arenas
<b>Journal:</b>	The R Journal
<b>Publisher:</b>	R FOUNDATION STATISTICAL COMPUTING
<b>DOI:</b>	10.32614/RJ-2022-044
<b>Year:</b>	2022
<b>Source of impact:</b>	WOS (JCR)
<b>Category:</b>	COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS
<b>Impact index:</b>	1.673 (Q4)
<b>Position:</b>	97/112



# dbcsp: User-friendly R package for Distance-Based Common Spatial Patterns

by *Itsaso Rodríguez, Itziar Irigoien, Basilio Sierra, and Concepción Arenas*

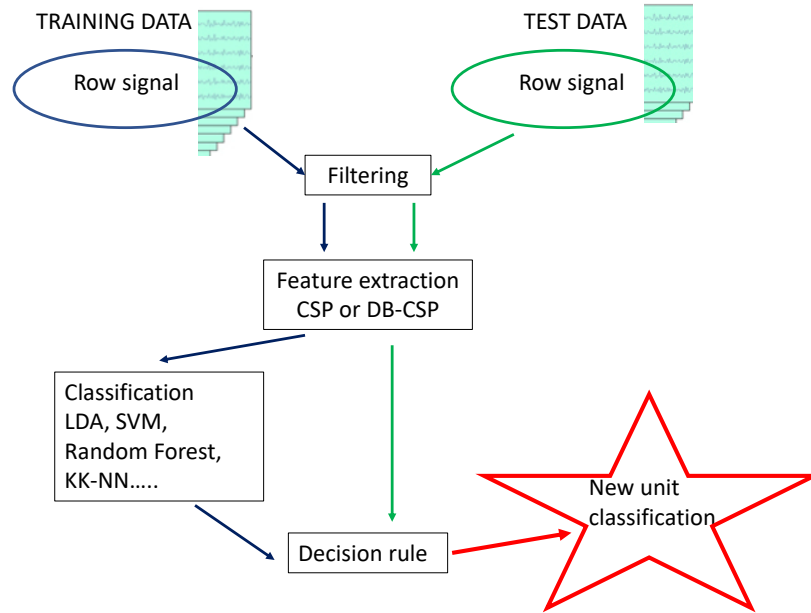
**Abstract** Common Spatial Patterns (CSP) is a widely used method to analyse electroencephalography (EEG) data, concerning the supervised classification of the activity of brain. More generally, it can be useful to distinguish between multivariate signals recorded during a time span for two different classes. CSP is based on the simultaneous diagonalization of the average covariance matrices of signals from both classes and it allows the data to be projected into a low-dimensional subspace. Once the data are represented in a low-dimensional subspace, a classification step must be carried out. The original CSP method is based on the Euclidean distance between signals, and here we extend it so that it can be applied on any appropriate distance for data at hand. Both the classical CSP and the new Distance-Based CSP (DB-CSP) are implemented in an R package, called **dbcsp**.

## 1 Background

Eigenvalue and generalized eigenvalue problems are very relevant techniques in data analysis. The well-known Principal Component Analysis with the eigenvalue problem in its roots was already established by the late seventies (Mardia et al., 1979). In mathematical terms, Common Spatial Patterns (CSP) is based on the generalized eigenvalue decomposition or the simultaneous diagonalization of two matrices to find projections in a low dimensional space. Although in algebraic terms PCA and CSP share several similarities, their main aims are different: PCA follows a non-supervised approach but CSP is a two-class supervised technique. Besides, PCA is suitable for standard quantitative data arranged in ‘individuals  $\times$  variables’ tables, while CSP is designed to handle multivariate signals time series. That means that, while for PCA each individual or unit is represented by a classical numerical vector, for CSP each individual is represented by several signals recorded during a time span, i.e., by a ‘number of signals  $\times$  time span’ matrix. CSP allows the individuals to be represented in a dimension reduced space, a crucial step given the high dimensional nature of the original data. CSP computes the average covariance matrices of signals from the two classes to yield features whose variances are optimal to discriminate the classes of measurements. Once data is projected into a low dimensional space, a classification step is carried out. The CSP technique was first proposed under the name Fukunaga-Koontz Transform in Fukunaga and Koontz (1970) as an extension of PCA, and Müller-Gerking et al. (1999) used it to discriminate electroencephalography data (EEG) in a movement task. Since then, it has been a widely used technique to analyze EEG data and develop Brain Computer Interfaces (BCI), with different variations and extensions (Blankertz et al., 2007a,b; Grosse-Wentrup and Buss, 2008; Lotte and Guan, 2011; Wang et al., 2012; Astigarraga et al., 2016; Darvish Ghanbar et al., 2021). In Wu et al. (2013), subject specific best time window and number of CSP features are fitted through a two-level cross validation scheme within the Linear Discriminant classifier. Samek et al. (2014) offer a divergence-based framework including several extensions of CSP. As a general term, CSP filter maximizes the variance of the filtered or projected EEG signals of one class of movements while minimizing it for the signals of the other class. Similarly, it can be used to detect epileptic activities Khalid et al. (2016) or other brain activities. BCI systems can also be of great help to people who suffer from some disorders of cerebral palsy, or who suffer from other diseases or disabilities that prevent the normal use of their motor skills. These systems can considerably improve the quality of life of these people, for which small advances and changes imply big improvements. BCI systems can also contribute to human vigilance detection, connected with occupations involving sustained attention tasks. Among others, CSP and variations of it have been applied to the vigilance estimation task (Yu et al., 2019).

The original CSP method is based on the Euclidean distance between signals. However, as far as we know, a generalization allowing the use of any appropriate distance was not introduced. The aim of the present work is to introduce a novel Distance-Based generalization of it (DB-CSP). This generalization is of great interest, since these techniques can also offer good solutions in other fields where multivariate time series data arise beyond pure electroencephalography data (Poppe, 2010; Rodríguez-Moreno et al., 2020).

Although CSP in its classical version is a very well-known technique in the field of BCI, it is not implemented in R. In addition, as DB-CSP is a new extension of it, it is worth building an R package that includes both CSP and DB-CSP techniques. The package offers functions in a user-friendly way for the less familiar users of R but it also offers complete information about its objects so that reproducible analysis can be carried out and more advanced and customised analysis can be performed taking



**Figure 1:** Flow-chart showing the steps to classify a new data. First, the filtering is done along with the feature extraction. This is the core of the procedure (CSP or DB-CSP). Then, a classifier is built to make the decision giving the classification of the new data.

advantage of already well-known packages of R.

The paper is organized as follows. First, we review the mathematical formulation of the Common Spatial Patterns method. Next, we present the core of our contribution describing both the novel CSP’ extension based on distances and the **dbcsp** package. Then, the main functions in **dbcsp** are introduced along with reproducible examples of their use. Finally, some conclusions are drawn.

## 2 CSP and DB-CSP

Let us consider that we have  $n$  statistical individuals or units classified in two classes  $C_1$  and  $C_2$ , with  $\#C_1 = n_1$  and  $\#C_2 = n_2$ . For each unit  $i$  in class  $C_k$ , data from  $c$  sources or signals are collected during  $T$  time units and therefore unit  $i$  is represented in matrix the  $X_{ik}$  ( $i = 1, \dots, n_k; k = 1, 2$ ). For instance, for electroencephalograms, data are recorded by a  $c$ -sensor cap each  $t$  time units ( $t = 1, \dots, T$ ). As usual, we consider that each  $X_{ik}$  is already scaled or with the appropriate pre-processing in the context of application; for instance, if working with EGG data, each signal should be band-pass filtered before its use.

The goal is to classify a new unit  $X$  in  $C_1$  or  $C_2$ . To this end, first a projection into a low-dimensional subspace is carried out. Then, as a standard approach the Linear Discriminant classifier (LDA) is applied taking as input data for the classifier the log-variance of the projections obtained in the first step. It is obvious that the importance of the technique lies mainly in the first step, and once it is done, LDA or any other classifiers could be applied. Based on that, we focus on how this projection into a low-dimensional space is done, from the classical CSP point of view as well as its novel extension DB-CSP (see Figure 1).

### Classical CSP

The main idea is to use a linear transform to project or filter data into low-dimensional subspace with a projection matrix, in such a way that each row consists of weights for signals. This transformation maximizes the variance of two-class signal matrices. The method performs a simultaneous diagonalization of the covariance matrices of both classes. Given data  $X_{11}, \dots, X_{n_1 1}$  (matrices  $c \times T$ ) from class  $C_1$  and  $X_{12}, \dots, X_{n_2 2}$  (also matrices  $c \times T$ ) from class  $C_2$ , the following steps are needed:



- All matrices are standardized so that traces of  $X_{ik}X'_{ik}$  are the same.
- Compute average covariance matrices:

$$B_k = \frac{1}{n_k} \sum_{i=1}^{n_k} X_{ik}X'_{ik}, \quad k = 1, 2$$

- Look for directions  $W = (\mathbf{w}_1, \dots, \mathbf{w}_c) \in \mathbb{R}^{c \times c}$  according to the criterion:

$$\begin{aligned} &\text{Maximize } tr(W' B_1 W) \\ &\text{subject to } W'(B_1 + B_2)W = I \end{aligned}$$

The solution is given by the generalized spectral decomposition  $B_1 \mathbf{w} = \lambda B_2 \mathbf{w}$  choosing the first and the last  $q$  eigenvectors:  $W_{CSP} = (\mathbf{w}_1, \dots, \mathbf{w}_q, \mathbf{w}_{c-q+1}, \dots, \mathbf{w}_c)$ .

Vectors  $\mathbf{w}_j$  offer weights so that new signals  $X'_{i1} \mathbf{w}_j$  and  $X'_{i2} \mathbf{w}_j$  have big and low variability for the first  $q$  vectors ( $j = 1, \dots, q$ ) respectively, and vice-versa for the last  $q$  vectors ( $j = c - q + 1, \dots, c$ ). To clarify the notation and interpretation, let us denote  $\mathbf{a}_j = \mathbf{w}_j$  the first  $q$  vectors and  $\mathbf{b}_j = \mathbf{w}_{c+1-j}$  the last  $q$ . That way, and broadly speaking, variability of elements in  $C_1$  is big when projecting on vectors  $\mathbf{a}_j$  and low on vectors  $\mathbf{b}_j$ , and vice-versa, for elements in class  $C_2$ .

Finally, the log-variability of these new and few  $2q$  signals are considered as input for the classification, which classically is the Linear Discriminant Analysis (LDA). Obviously, any other classification technique can be used, as it is illustrated in the subsection **Extending the example**.

### Distance-based CSP

Following the commented ideas, the Distance-Based CSP (DB-CSP) is an extension of the classical CSP method. In the same way as the classical CSP, DB-CSP gives some weights to the original sources or signals and obtains new and few  $2q$  signals which are useful for the discrimination between the two classes. Nevertheless, the considered distance between the signals can be any other than the Euclidean. The steps are the following:

- Compute an appropriate distance measure between sources and the double-centered inner product:

$$X_{ik} \rightarrow D_{ik} \rightarrow P_{ik} = -1/2HD_{ik}^{(2)}H, \quad i = 1, \dots, n_k; k = 1, 2$$

where  $H$  stands for the centering matrix and the superindex in brackets (2) for squared elements in the matrix. Again, all matrices are standardized so that all traces of  $X_{ik}X'_{ik}$  are the same.

- Compute average distance-based covariance matrices:

$$B_k^* = \frac{1}{n_k} \sum_{i=1}^{n_k} (P_{ik}P'_{ik} + X_{ik}\mathbf{x}_{ik}\mathbf{1}' + \mathbf{1}\mathbf{x}'_{i,k}X'_{ik} - \mathbf{x}'_{ik}\mathbf{x}_{ik}\mathbf{1}\mathbf{1}')$$

where  $\mathbf{x}_{ik} = \frac{1}{c}\mathbf{1}'X_{ik}$ , and  $k = 1, 2$ .

Once we have the covariance matrices related to the chosen distance matrix, the directions are found as in classical CSP and new signals  $X'_{i1}\mathbf{a}_j$ ,  $X'_{i2}\mathbf{b}_j$  are built ( $j = 1, \dots, q$ ). Again, for individuals in class  $C_1$  the projections on vectors  $\mathbf{a}$  and  $\mathbf{b}$  are big and low respectively; for individuals in class  $C_2$  it is the other way round.

It is important to note that if the chosen distance does not produce a positive definite covariance matrix, it must be replaced by a similar one that is positive definite.

When the selected distance is the Euclidean, then, **DB-CSP reduces to classical CSP**.

Once the  $q$  directions  $\mathbf{a}_j$  and  $\mathbf{b}_j$  are calculated, new  $2q$  signals are built. Many interesting characteristics of the new signals could be extracted, although the most important in the procedure is the variance. Those characteristics of the new signals are the input data for the classification step.

### 3 Implementation

In this section, the structure of the package and the functions implemented are explained. The **dbcsp** package was developed for the free statistical R environment and it is available from the Comprehensive R Archive Network (CRAN) at <https://cran.r-project.org/web/packages/dbcsp/index.html>.

#### Input

The input data are the corresponding  $n_1$  and  $n_2$  matrices  $X_{ik}$  of the  $n$  units classified in classes  $C_1$  and  $C_2$ , respectively ( $i = 1, \dots, n_k$ ;  $k = 1, 2$ ). Let  $x_1$  and  $x_2$  be two lists of length  $n_1$  and  $n_2$ , respectively, with  $X_{ik}$  matrices ( $c \times T$ ) as elements of the lists. NA values are allowed. They are imputed by interpolating with the surrounding values via the `na.approx` function in package **zoo**. To ensure the user is aware of the missing values and their imputation, a warning is printed. We also consider that new items to be classified are in list  $x_t$ . The aforementioned first step of the method is carried out by building the object called "dbcsp".

#### dbcsp object

The `dbcsp` object is an S4 class created to compute the projection vectors  $W$ . The object has the following slots:

- **Slots**

$X_1 = \text{"list"}, X_2 = \text{"list"}$ , the lists  $X_1$  and  $X_2$  (lengths  $n_1$  and  $n_2$ ) containing the matrices  $X_{ik}$  for the two classes  $C_1$  and  $C_2$ , respectively ( $i = 1, \dots, n_k$ ;  $k = 1, 2$ ).

$q = \text{"integer"}$ , to determine the number of pairs of eigenvectors  $\mathbf{a}_j$  and  $\mathbf{b}_j$  that are kept. By default  $q=15$ .

$labels = \text{"character"}$ , vector of two strings indicating labels names, by default names of elements in  $X_1$  and  $X_2$ .

$type = \text{"character"}$ , to set the type of distance to be considered, by default  $type='EUCL'$ . The supported distances are these ones:

- Included in **TSdist**: `infnorm, ccor, sts, ...`
- Included in **parallelDist**: `bhattacharyya, bray, ...`
- Custom distances: it is also possible to use a user-defined distance function, a function `dcustom` which returns a scalar providing the distance value ( $d(x_{ik}, x_{jk})$ ) between signals  $x_{ik}$  and  $x_{jk}$  ( $i, j = 1, \dots, n_k$ ,  $k = 1, 2$ ). The name of the custom distance function is passed as character to the type parameter ( $type="dcustom"$ ). The `parallelDist` package also allows the use of custom distances, but the distance function has to be defined using the `cppXPtr` function of the **RcppXPTrUtils** package, as is explained in the *User-defined distance functions* section of the `parallelDist` package documentation.

$mixture = \text{"logical"}$ , logical value indicating whether to use mixture of distances or not (EUCL + other), by default  $mixture=FALSE$ .

$w = \text{"numeric"}$ , weight for the mixture of distances  $D_{mixture} = wD_{euclidean} + (1 - w)D_{type}$ , by default  $w=0.5$ .

$training = \text{"logical"}$ , logical value indicating whether or not to perform the classification, by default  $classification=FALSE$ . If  $classification=TRUE$ , LDA discrimination based on the log-variances of the projected sources is considered, following the classical approach in CSP.

$fold = \text{"integer"}$ , integer value, by default  $fold=10$ . It controls the number of partitions for the  $k$ -fold validation procedure, if the classification is done.

$seed = \text{"numeric"}$ , numeric value, by default  $seed=NULL$ . Set a seed in case you want to be able to replicate the results.

`eig.tol = "numeric"`, numeric value, by default `eig.tol=1e-06`. If the minimum eigenvalue is below this tolerance, average covariance matrices are replaced by the most similar matrix that is positive definite. It is done via function `nearPD` in **Matrix** and a warning message is printed to make the user aware of it.

`out = "list"`, list containing elements of the output. Mainly, matrix  $W$  with vectors  $\mathbf{a}_j$  and  $\mathbf{b}_j$  in element vectors, log-variances of filtered signals in `proy` and partitions considered in the  $k$ -fold approach with reproducibility purposes.

- **Usage**

Following the standard procedure in R, an instance of a class `dbcsp` is created via the `new()` constructor function:

```
new("dbcsp", X1 = x1, X2 = x2)
```

Slots `X1` and `X2` are compulsory since they contain the original data. When a slot is not specified, the default value is considered. First, the S4 object of class `dbcsp` must be created. By default, the Euclidean distance is used, nevertheless it can be changed. For instance, "Dynamic Transform Distance" (Giorgino et al., 2009) can be set:

```
mydbcsp <- new('dbcsp', X1=x1, X2=x2, type='dtw')
```

or a mixture between this distance and the Euclidean can be indicated by:

```
mydbcsp.mix <- new('dbcsp', X1=x1, X2=x2, labels=c("C1", "C2"),
  mixture=TRUE, w=0.4, type="dtw")
```

Besides, a custom distance function can be defined and used when creating the object:

```
fn <- function(x, y) mean(1 - cos(x - y))
mydbcsp <- new("dbcsp", X1 = x, X2 = y, type="fn")
```

It is worth mentioning that it is possible to reduce the computational time through `parallelDist` custom distance option, where the function is defined using C++ and by creating an external pointer to the function by means of the `cppXPtr` function:

```
customEucli <- RcppXPTrUtils::cppXPTr(
  "double customDist(const arma::mat &A, const arma::mat &B) {
    return sqrt(arma::accu(arma::square(A - B)));
  }",
  depends = c("RcppArmadillo")
)
mydbcsp <- new('dbcsp', x1, x2, type="customEucli")
```

The object contains all the information to carry out the classification task in a lower dimension space.

## Functions plot and boxplot

For exploratory and descriptive purposes, the original signals  $X_{ik}$  and the projected ones can be plotted for the selected individual  $i$  in class  $k$ , and the selected pair of dimensions  $\mathbf{a}_j$  and  $\mathbf{b}_j$  ( $i = 1, \dots, n_k$ ,  $k = 1, 2$ ).

- **Usage**

```
plot(mydbcsp)
```

- **Arguments**

`x`, an object of class `dbcsp`

`class`, integer to indicate which of both classes to access (1 or 2), by default `class=1`.

`index`, integer to indicate which instance of the class to plot, by default `index=1`.

vectors, integer to indicate which  $j$  projected signals are to be plotted. By default all the vectors used in the projection are plotted.

pairs logical, if TRUE the pairs  $\mathbf{a}_j$  and  $\mathbf{b}_j$  of the indicated indices are also shown, by default pairs=TRUE.

before logical, if TRUE the original signals are plotted, by default before=TRUE.

after logical, if TRUE the signals after projection are plotted, by default after=TRUE.

legend logical, if TRUE, a legend for filtered signals is shown, by default legend=FALSE.

getsignals logical, if TRUE, the projected signals are returned.

Besides, the log-variances of the projected signals of both classes can be shown in boxplots. This graphic can help to understand the discriminative power that is in the low-dimension space.

- **Usage**

`boxplot(mydbcsp)`

- **Arguments**

`x`, an object of class `dbcsp`

vectors, integer or vector of integers, indicating the index of the projected vectors to plot, by default `index=1`.

pairs logical, if TRUE the pairs  $\mathbf{a}_j$  and  $\mathbf{b}_j$  of the indicated indices are also shown, by default pairs=TRUE.

show\_log logical, if TRUE the logarithms of the variances are displayed, otherwise the variances, by default `show_log=TRUE`.

It is worth taking into account that in the aforementioned functions, values in argument vectors must lie between 1 and  $2q$ , being  $q$  the number of dimensions used to perform the DB-CSP algorithm when creating the `dbcsp` object. Therefore, values 1 to  $q$  correspond to vectors  $\mathbf{a}_1$  to  $\mathbf{a}_q$  and values  $q + 1$  to  $2q$  correspond to vectors  $\mathbf{b}_1$  to  $\mathbf{b}_q$ . Then, if pairs=TRUE, it is recommended that values in argument vectors are in  $\{1, \dots, q\}$ , since their pairs are plotted as well. When values are above  $q$ , it should be noted that they correspond to vectors  $\mathbf{b}_1$  to  $\mathbf{b}_q$ . For instance, if  $q=15$  and `boxplot(object, vectors=16, pairs=FALSE)`, vector  $\mathbf{b}_1$  ( $16 - q = 1$ ) is shown.

### Function `selectQ`, Function `train` and Function `predict`

The functions in this section help the classification step in the procedure. Function `selectQ` helps to find an appropriate dimension needed for the classification. Given different values of dimensions, the accuracy related to each dimension is calculated so that the user can assess which dimension of the reduced space can be sufficient. A  $k$ -fold cross-validation approach or a holdout approach can be followed. Function `train` performs the Linear Discriminant classification based on the log-variances of the dimensions built in the `dbcsp` object. Since LDA has a geometric interpretation that makes the classifier sensible for more general situations [Duda et al. \(2001\)](#), not the normality nor the homoscedasticity of data are checked. The accuracy of the classifier is computed based on the  $k$ -fold validation procedure. Finally, function `predict` performs the classification of new individuals.

- **Usage of `selectQ`**

`selectQ(mydbcsp)`

- **Arguments**

`object`, an object of class `dbcsp`

`Q`, vector of integers which represents the dimensions to use, by default `Q=c(1, 2, 3, 5, 10, 15)`.

`train_size`, float between 0.0 and 1.0 representing the proportion of the data set to include in the train split, by default `train_size=0.75`.

`CV`, logical indicating whether a  $k$ -fold cross validation must be performed or a hold-out approach (if TRUE, `train_size` is not used), by default `CV=FALSE`.

`folds` integer, number of folds to use if CV is performed.

`seed` numeric value, by default `seed=NULL`. Set a seed in case you want to be able to replicate the results.

This function returns the accuracy values related to each dimension set in `Q`. If `CV=TRUE`, the mean accuracy as well as the standard deviation among folds is also returned.

- **Usage of train**

`train(mydbcsp)` or embedded as a parameter in:  
`new('dbcsp', X1=x1, X2=x2, training=TRUE, type="dtw")`

- **Arguments**

`x`, an object of class `dbcsp`

`selected_q`, integer value indicating the number of vectors to use when training the model. By default all dimensions considered when creating the object `dbcsp`.

Besides, arguments `seed` and `fold` are available.

It is important to note that in this way a classical analysis can be carried out, in the sense of:

- LDA is applied based on the log-variances of the dimensions indicated by the user in `select_q`;
- percentage of correct classification is obtained via  $k$ -fold cross validation.

However, it is evident that it may be of interest to use other classifiers or other characteristics in addition to or different from log-variances. This more advanced procedure is explained below. See the basic analysis of the **User guide with a real example** section in order to visualize and follow the process of a first basic/classic analysis.

- **Usage of predict**

`predict(mydbcsp, X_test=xt)`

- **Arguments**

`object`, an object of class `dbcsp`

`X_test`, list of matrices to be classified.

`true_targets`, optional, if available, vector of true labels of the instances. Note that they must match the name of the labels used when training the model.

## 4 User guide with a real example

To show an example beyond pure electroencephalography data, Action Recognition data is considered. Besides having a reproducible example to show the use of the implemented functions and the results they offer, this Action Recognition data set is included in the package. The data set contains the skeleton data extracted from videos of people performing six different actions, recorded by a semi-humanoid robot. It consists of a total of 272 videos with 6 action categories. There are around 45 clips in each category, performed by 46 different people. Each instance is composed of 50 signals ( $xy$  coordinates for 25 body key points extracted using OpenPose (Cao et al., 2019)), where each signal has 92 values, one per frame. These are the six categories included in the data set:

1. Come: gesture for telling the robot to come to you. There are 46 instances for this class.
2. Five: gesture of 'high five'. There are 45 instances for this class.
3. Handshake: gesture of handshaking with the robot. There are 45 instances for this class.
4. Hello: gesture for saying hello to the robot. There are 44 instances for this class.
5. Ignore: ignore the robot, pass by. There are 46 instances for this class.
6. Look at: stare at the robot in front of it. There are 46 instances for this class.

The data set is accessible via `AR.data` and more specific information can be found in (Rodríguez-Moreno et al., 2020). Each class is a list of matrices of  $[K \times \text{num\_frames}]$  dimensions, where  $K = 50$  signals and  $\text{num\_frames} = 92$  values. As mentioned before, the 50 signals represent the  $xy$  coordinates of 25 body key points extracted by OpenPose.

For example, two different classes can be accessed this way:

```
x1 <- AR.data$come
x2 <- AR.data$five
```

where, `x1` is a list of 46 instances of  $[50 \times 92]$  matrices of *come* class and `x2` is a list of 45 instances of  $[50 \times 92]$  matrices of *five* class. An example of skeleton sequences for both classes is shown in Figure 2 (left, for class *come* and right, for class *five*).



**Figure 2:** Sequences of the skeleton extracted from the videos. Left: sequence for action 'come'. Right: sequence for action '(high) five'. For each frame,  $x$  and  $y$  coordinates of the 25 body key points of the skeleton are extracted by OpenPose.

Next, the use of functions in `dbcsp` is shown based on this data set. First a basic/classic analysis is performed.

### Basic/classic analysis

Let us consider an analysis using 15-dimensional projections and the Euclidean distance. At a first step the user can obtain vectors  $W$  by:

```
x1 <- AR.data$come
x2 <- AR.data$five
mydbcsp <- new('dbcsp', X1=x1, X2=x2, q=15, labels=c("C1", "C2"))
summary(mydbcsp)
```

Creating the object `mydbcsp`, the vectors  $W$  are calculated. As indicated in parameter  $q=15$ , the first and last 15 eigenvectors are retained. With `summary`, the obtained output is:

```
There are 46 instances of class C1 with [50x92] dimension.
There are 45 instances of class C2 with [50x92] dimension.
The DB-CSP method has used 15 vectors for the projection.
EUCL distance has been used.
Training has not been performed yet.
```

Now, if the user knows from the beginning that 3 is an appropriate dimension, the classification step could be done while creating the object. Using classical analysis, with for instance 10-fold, LDA as classifier and log-variances as characteristics, the corresponding input and summary output are:

```
mydbcsp <- new('dbcsp', X1=x1, X2=x2, q=3, labels=c("C1", "C2"), training=TRUE, fold = 10, seed = 19)
summary(mydbcsp)
```

There are 46 instances of class C1 with [50x92] dimension.  
 There are 45 instances of class C2 with [50x92] dimension.  
 The DB-CSP method has used 3 vectors for the projection.  
 EUCL distance has been used.  
 An accuracy of 0.9130556 has been obtained with 10 fold cross validation and using 3 vectors when training.

If a closer view of the accuracies among the folds is needed, the user can obtain them from the out slot of the object:

```
# Accuracy in each fold
mydbcsp@out$folds_acc

# Intances belonging to each fold
mydbcsp@out$used_folds
```

### Basic/classic analysis selecting the value of $q$

Furthermore, it is clear that the optimal value of  $q$  should be chosen based on the percentages of correct classification. It is worth mentioning that the LDA is applied on the  $2q$  projections, as set in the object building step. It is interesting to measure how many dimensions would be enough using selectQ function:

```
mydbcsp <- new('dbcsp', X1=x1, X2=x2, labels=c("C1", "C2"))
selectDim <- selectQ(mydbcsp, seed=30, CV=TRUE, fold = 10)
```

```
selectDim
  Q   acc   sd
1  1 0.7663889 0.12607868
2  2 0.9033333 0.09428818
3  3 0.8686111 0.11314534
4  5 0.8750000 0.13289537
5 10 0.8797222 0.09513230
6 15 0.8250000 0.05257433
```

Since the 10-fold cross-validation approach is chosen, the mean accuracies as well as the corresponding standard deviations are returned. Thus, with Linear Discriminant Analysis (LDA), log-variances as characteristics, it seems that dimensions related to first and last  $q = 2$  eigenvectors ( $2 \times 2$  dimensions in total) are enough to obtain a good classification, with an accuracy of 90%. Nevertheless, it can also be observed that variation among folds can be relevant.

To visualize what is the representation in the reduced dimension space function plot can be used. For instance, to visualize the first unit of the first class, based on projections along the 2 first and last vectors ( $\mathbf{a}_1, \mathbf{a}_2$  and  $\mathbf{b}_1, \mathbf{b}_2$ ):

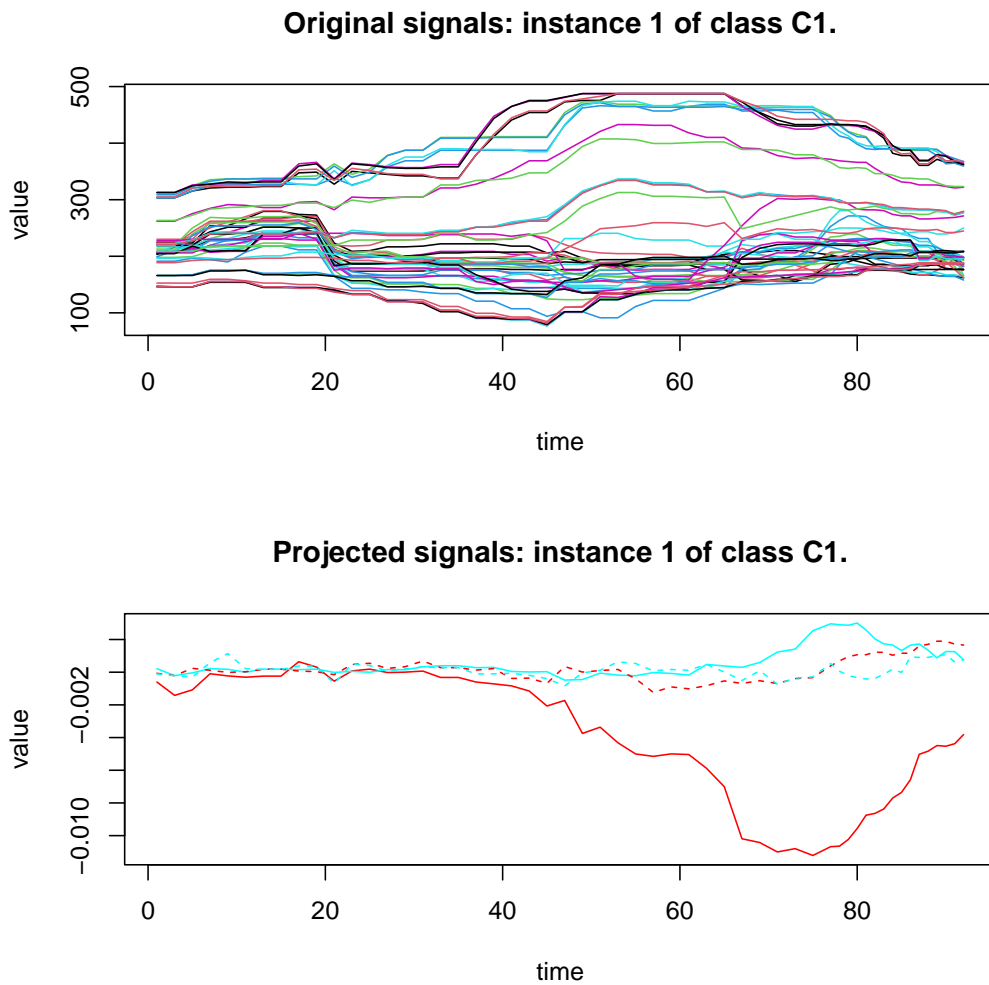
```
plot(mydbcsp, index=1, class=1, vectors=1:2)
```

In the top graphic of Figure 3, the representation of the first video of class  $C_1$  given by non standardized matrix  $X_{11}$  can be seen, where the horizontal axis represents the frames of the video and the lines are the positions of the body key points (50 lines). In the bottom graphic, the same video is represented in a reduced space where the video is represented by the new signals (only 4 lines).

To have a better insight of the discriminating power of the new signals in the reduced dimension space, we can plot the corresponding log-variances of the new signals. Parameter vectors in function boxplot sets which are the eigenvectors considered to plot.

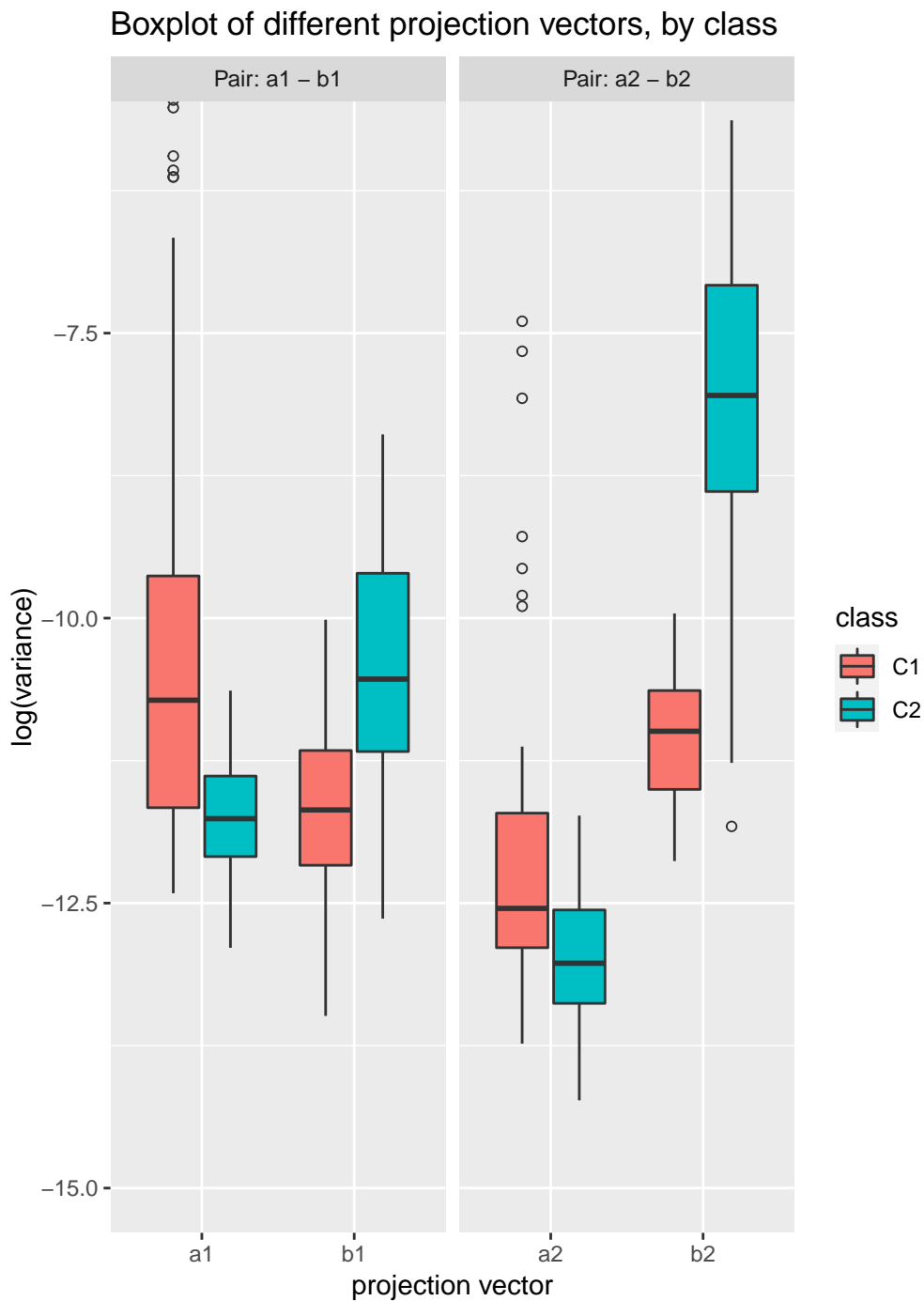
```
boxplot(mydbcsp, vectors=1:2)
```

In Figure 4 it can be seen that variability of projections on the first eigenvector direction ( $\log(\text{VAR}(X'_{ik}\mathbf{a}_1)))$  are big for elements in  $x_1$ , but small for elements in  $x_2$ . Analogously, projecting on the last dimension ( $\log(\text{VAR}(X'_{ik}\mathbf{b}_1)))$ , low variability is held in  $x_1$  and big variability in  $x_2$ . The same pattern holds when projecting on vectors  $\mathbf{a}_2$  and  $\mathbf{b}_2$ .



**Figure 3:** Representation of the first video of class  $C_1$ . Top: original version where each line corresponds to the signal of a body key point. Bottom: the projections on vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  (continuous lines) and  $\mathbf{b}_1$  and  $\mathbf{b}_2$  (dotted lines). Being a video of class  $C_1$ , variabilities of the projections on vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  are big whereas on vectors  $\mathbf{b}_1$  and  $\mathbf{b}_2$  are small, as expected.





**Figure 4:** Log-variabilities of the projected signals on vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  and  $\mathbf{b}_1$  and  $\mathbf{b}_2$ , separated by classes  $C_1$  and  $C_2$ . By construction, variabilities of the projections on vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$  are big for units in class  $C_1$  and small for units  $C_2$ ; opposite pattern can be seen for projections on vectors  $\mathbf{b}_1$  and  $\mathbf{b}_2$ .

### Basic/classic analysis new unit classification

Once the value of  $q$  has been decided and the accuracy of the classification is known, the classifier should be built (through `train()`) so that the user can proceed to predict the class a new action held in a video belongs to, using the function `predict`. For instance, with only illustrative purpose, we can classify the first 5 videos which are stored in `x1`.

```
mydbcsp <- train(mydbcsp, selected_q=2, verbose=FALSE)
xtest <- x1[1:5]
outpred <- predict(mydbcsp, X_test=xtest)
```

If the labels of the testing items are known, the latter function returns the accuracy.

```
outpred <- predict(mydbcsp, X_test=xtest, true_targets= rep("C1", 5))
```

Finally, notice that the user could use any other distance instead of the Euclidean between the signals to compute the important directions  $\mathbf{a}_j$  and  $\mathbf{b}_j$ . For instance, in this case it could be appropriate to use the Dynamic Time Warping distance, setting so in the argument `type="dtw"`:

```
# Distance DTW
mydbcsp.dtw <- new('dbcsp', X1=x1, X2=x2, labels=c("C1", "C2"), type="dtw")
```

## 5 Extending the example

In the previous section a basic workflow to use functions implemented in `dbcsp` is presented. Nevertheless, it is straightforward to extend the procedure. Once the interesting directions in  $W$  are calculated through `dbcsp`, other summarizing characteristics beyond the variance could be extracted from the projected signals, as well as other classifiers which could be used in the classification step. For those purposes, `dbcsp` is used to compute the directions in  $W$  that will be the base to calculate other features as well as the input features for other classifiers. Here it is shown how, once the eigenvectors are extracted from an object `dbcsp`, several characteristics could be extracted from the signals and a new data.frame can be built so that any other classification technique could be applied. In this example we worked with `caret` package to apply different classifiers. It is important to pay attention to which the train and test sets are, so that the vectors are computed based only on training set instances.

```
# Establish training and test data
n1 <- length(x1)
trainind1 <- rep(TRUE, n1)
n2 <- length(x2)
trainind2 <- rep(TRUE, n2)
set.seed(19)
trainind1[sample(1:n1, 10, replace=FALSE)] <- FALSE
trainind2[sample(1:n2, 10, replace=FALSE)] <- FALSE
x1train <- x1[trainind1]
x2train <- x2[trainind2]

# Extract the interesting directions
vectors <- new('dbcsp', X1=x1train, X2=x2train, q=5, labels=c("C1", "C2"))@out$vectors

# Function to calculate the desired characteristics from signals
calc_info <- function(proj_X, type){
  values <- switch(type,
    'var' = values <- plyr::laply(proj_X, function(x){apply(x,1,var)}),
    'max' = values <- plyr::laply(proj_X, function(x){apply(x,1,max)}),
    'min' = values <- plyr::laply(proj_X, function(x){apply(x,1,min)}),
    'iqr' = values <- plyr::laply(proj_X, function(x){
      apply(x,1,function(y){
        q <- quantile(y, probs = c(0.25, 0.75))
        q[2] -q[1]
      })
    })
  )
  return(values)
}
```

By means of this latter function, besides the variance of the new signals, the maximum, the minimum, and the interquartile range can be extracted.

Next, imagine we want to perform our classification step with the interquartile range information along with the log-variance.

```
# Project units of class C1 and
projected_x1 <- plyr::llply(x1, function(x,W) t(W)%*%x, W=vectors)

# Extract the characteristics
logvar_x1 <- log(calc_info(projected_x1,'var'))
iqr_x1 <- calc_info(projected_x1,'iqr')
new_x1 <- data.frame(logvar=logvar_x1, iqr=iqr_x1)

# Similarly for units of class C2
projected_x2 <- plyr::llply(x2, function(x,W) t(W)%*%x, W=vectors)
logvar_x2 <- log(calc_info(projected_x2,'var'))
iqr_x2 <- calc_info(projected_x2,'iqr')
new_x2 <- data.frame(logvar=logvar_x2, iqr=iqr_x2)

# Create dataset for classification
labels <- rep(c('C1','C2'), times=c(n1,n2))
new_data <- rbind(new_x1,new_x2)
new_data$label <- factor(labels)
new_data_train <- new_data[c(trainind1, trainind2), ]
new_data_test <- new_data[!c(trainind1, trainind2), ]

# Random forest
trControl <- caret::trainControl(method = "none")
rf_default <- caret::train(label~.,
                           data = new_data_train,
                           method = "rf",
                           metric = "Accuracy",
                           trControl = trControl)

rf_default

# K-NN
knn_default <- caret::train(label~.,
                            data = new_data_train,
                            method = "knn",
                            metric = "Accuracy",
                            trControl = trControl)

knn_default

# Predictions and accuracies on test data
# Based on random forest classifier
pred_labels <- predict(rf_default, new_data_test)
predictions_rf <- caret::confusionMatrix(table(pred_labels,new_data_test$label))
predictions_rf

# Based on knn classifier
pred_labels <- predict(knn_default, new_data_test)
predictions_knn <- caret::confusionMatrix(table(pred_labels,new_data_test$label))
predictions_knn
```

Thus, it is easy to integrate results and objects that **dbcsp** builds so that they can be integrated with other R packages and functions. This is interesting for more advanced users to perform their own customized analysis.

## 6 Conclusions

In this work a new Distance-Based Common Spatial Pattern is introduced. It allows to perform the classical Common Spatial Pattern when the Euclidean distance between signals is considered, but

it can be extended to the use of any other appropriate distance between signals as well. All of it is included in package the **dbcsp**. The package is easy to use for non-specialised users but, for the sake of flexibility, more advanced analysis can be carried out combining the created object and obtained results with already well-known R packages, such as **caret**, for instance.

## Acknowledgements

This research was partially supported: IR by The Spanish Ministry of Science, Innovation and Universities (FPU18/04737 predoctoral grant). II by the Spanish Ministerio de Economía y Competitividad (RTI2018-093337-B-I00; PID2019-106942RB-C31). CA by the Spanish Ministerio de Economía y Competitividad (RTI2018-093337-B-I00, RTI2018-100968-B-I00) and by Grant 2017SGR622 (GRBIO) from the Departament d'Economia i Coneixement de la Generalitat de Catalunya. BS II by the Spanish Ministerio de Economía y Competitividad (RTI2018-093337-B-I00).

## Author's contributions

II and CA designed the study. IR and II wrote and debugged the software. IR, II and CA checked the software. II, CA, IR and BS wrote and reviewed the manuscript. All authors have read and approved the final manuscript.

## Bibliography

- A. Astigarraga, A. Arruti, J. Muguerza, R. Santana, J. I. Martin, and B. Sierra. User adapted motor-imaginary brain-computer interface by means of EEG channel selection based on estimation of distributed algorithms. *Mathematical Problems in Engineering*, page 1435321, 2016. URL <https://doi.org/10.1155/2016/1435321>. [p80]
- B. Blankertz, M. Kawanabe, R. Tomioka, F. U. Hohlefeld, V. V. Nikulin, and K.-R. Müller. Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing. In *NIPS'07: Proceedings of the 20th International Conference on Neural Information Processing*, pages 113–120, 2007a. [p80]
- B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Müller. Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine*, 25(1):41–56, 2007b. URL <https://doi.org/10.1109/MSP.2008.4408441>. [p80]
- Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. OpenPose: realtime multi-person 2D pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2019. URL <https://doi.org/10.1109/TPAMI.2019.2929257>. [p86]
- K. Darvish Ghanbar, T. Yousefi Rezaii, A. Farzamnia, and I. Saad. Correlation-based common spatial pattern (CCSP): A novel extension of CSP for classification of motor imagery signal. *PLOS ONE*, 16:1–18, 2021. doi: 10.1371/journal.pone.0248511. URL <https://doi.org/10.1371/journal.pone.0248511>. [p80]
- R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2001. [p85]
- K. Fukunaga and W. L. Koontz. Application of the Karhunen-Loève expansion to feature selection and ordering. *IEEE Transactions on Computers*, 100(4):311–318, 1970. [p80]
- T. Giorgino et al. Computing and visualizing dynamic time warping alignments in R: the dtw package. *Journal of statistical Software*, 31(7):1–24, 2009. URL <http://dx.doi.org/10.18637/jss.v031.i07>. [p84]
- M. Grosse-Wentrup and M. Buss. Multiclass common spatial patterns and information theoretic feature extraction. *IEEE Transactions on Biomedical Engineering*, 55(8):1991–2000, 2008. URL <https://doi.org/10.1109/TBME.2008.921154>. [p80]
- M. I. Khalid, T. Alotaiby, S. A. Aldosari, S. A. Alshebeili, M. H. Al-Hameed, F. S. Y. Almohammed, and T. S. Alotaibi. Epileptic MEG spikes detection using common spatial patterns and linear discriminant analysis. *IEEE Access*, 4:4629–4634, 2016. URL <https://doi.org/10.1109/access.2016.2602354>. [p80]

- F. Lotte and C. Guan. Regularizing common spatial patterns to improve BCI designs: Unified theory and new algorithms. *Transactions on Biomedical Engineering*, 58(2):355–362, 2011. URL <https://doi.org/10.1109/TBME.2010.2082539>. [p80]
- K. Mardia, J. Kent, and J. Bibby. *Multivariate Analysis*. Academic Press, London, 1979. [p80]
- J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg. Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clinical Neurophysiology*, 110(5):787–798, 1999. URL [https://doi.org/10.1016/S1388-2457\(98\)00038-8](https://doi.org/10.1016/S1388-2457(98)00038-8). [p80]
- R. Poppe. Common spatial patterns for real-time classification of human actions. In *Machine Learning for Human Motion Analysis: Theory and Practice*, pages 55–73. IGI Global, 2010. [p80]
- I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, I. Rodríguez-Rodríguez, and B. Sierra. Shedding light on people action recognition in social robotics by means of common spatial patterns. *Sensors*, 20(8):2436, 2020. [p87]
- I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra, I. Irigoien, I. Rodríguez-Rodríguez, and I. Goienetxea. Using common spatial patterns to select relevant pixels for video activity recognition. *Applied Sciences*, 10(22), 2020. URL <https://www.mdpi.com/2076-3417/10/22/8075>. [p80]
- W. Samek, M. Kawanabe, and K.-R. Müller. Divergence-based framework for common spatial patterns algorithms. *IEEE Reviews in Biomedical Engineering*, 7:50–72, 2014. URL <https://doi.org/10.1109/RBME.2013.2290621>. [p80]
- H. Wang, Q. Tang, and W. Zheng. L1-norm-based common spatial patterns. *IEEE Transactions on Biomedical Engineering*, 59(3):653–662, 2012. URL <https://doi.org/10.1109/TBME.2011.2177523>. [p80]
- S.-L. Wu, C.-W. Wu, N. R. Pal, C.-Y. Chen, S.-A. Chen, and C.-T. Lin. Common spatial pattern and linear discriminant analysis for motor imagery classification. In *2013 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*, pages 146–151. IEEE, 2013. [p80]
- H. Yu, H. Lu, S. Wang, K. Xia, Y. Jiang, and P. Qian. A general common spatial patterns for EEG analysis with applications to vigilance detection. *IEEE Access*, 7:111102–111114, 2019. URL <https://doi.org/10.1109/ACCESS.2019.2934519>. [p80]

*Itsaso Rodríguez*

*Department of Computation Science and Artificial Intelligence, University of the Basque Country UPV/EHU  
Manuel Lardizabal 1, Donostia*

*Spain*

[itsaso.rodriguez@ehu.es](mailto:itsaso.rodriguez@ehu.es)

*Itziar Irigoien*

*Department of Computation Science and Artificial Intelligence, University of the Basque Country UPV/EHU  
Manuel Lardizabal 1, Donostia*

*Spain*

[itziar.irigoien@ehu.es](mailto:itziar.irigoien@ehu.es)

*Basilio Sierra*

*Department of Computation Science and Artificial Intelligence, University of the Basque Country UPV/EHU  
Manuel Lardizabal 1, Donostia*

*Spain*

[b.sierra@ehu.es](mailto:b.sierra@ehu.es)

*Concepción Arenas*

*Department of Genetics, Microbiology and Statistics. Statistics Section, University of Barcelona UB  
Diagonal 645, Barcelona*

*Spain*

[carenas@ub.edu](mailto:carenas@ub.edu)



## Video Activity Recognition: State-of-the-Art

<b>Title:</b>	Video Activity Recognition: State-of-the-Art
<b>Authors:</b>	I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra, I. Rodríguez, E. Jauregi
<b>Journal:</b>	Sensors
<b>Publisher:</b>	MDPI
<b>DOI:</b>	10.3390/s19143160
<b>Year:</b>	2019
<b>Times cited:</b>	60 (Google Scholar) / 45 (Scopus)
<b>Source of impact:</b>	WOS (JCR)
<b>Category:</b>	ENGINEERING, ELECTRICAL & ELECTRONIC
<b>Impact index:</b>	3.275 (Q2)
<b>Position:</b>	77/266





Review

# Video Activity Recognition: State-of-the-Art

Itsaso Rodríguez-Moreno <sup>1,\*</sup> , José María Martínez-Otzeta <sup>1</sup>, Basilio Sierra <sup>1</sup>, Igor Rodriguez <sup>1</sup> and Ekaitz Jauregi <sup>2</sup>

<sup>1</sup> Department of Computer Science and Artificial Intelligence, University of the Basque Country, Manuel Lardizabal 1, 20018 Donostia-San Sebastián, Spain

<sup>2</sup> Department of Computer Languages and Systems, University of the Basque Country, Manuel Lardizabal 1, 20018 Donostia-San Sebastián, Spain

\* Correspondence: itsaso.rodriguez@ehu.eus; Tel.: +34-943-015-107

Received: 13 June 2019; Accepted: 9 July 2019; Published: 18 July 2019



**Abstract:** Video activity recognition, although being an emerging task, has been the subject of important research efforts due to the importance of its everyday applications. Surveillance by video cameras could benefit greatly by advances in this field. In the area of robotics, the tasks of autonomous navigation or social interaction could also take advantage of the knowledge extracted from live video recording. The aim of this paper is to survey the state-of-the-art techniques for video activity recognition while at the same time mentioning other techniques used for the same task that the research community has known for several years. For each of the analyzed methods, its contribution over previous works and the proposed approach performance are discussed.

**Keywords:** activity recognition; computer vision; optical flow; deep learning

## 1. Introduction

Activity recognition consists of identifying some actions from a series of observations. This field has caught the interest of many researchers since the 1980s due to the number of applications for which it is useful, such as medicine [1,2], human–computer interaction [3,4], surveillance [5,6] or sociology [7,8]. For instance, in surveillance [9,10], the automatic detection of suspicious actions would allow for launching a warning and taking measures against any danger. Another example is the use of activity recognition for rehabilitation [11], recognizing the action the patients are performing and having the ability to determine if it is right or not. One of the main techniques used for activity recognition is computer vision, namely video-based activity recognition. Visual video features provide basic information for video events or actions.

The task of tracking and understanding what is happening in a video can be very challenging. Many attempts have been made lately using different techniques [12–14] such as optical flow [15,16], Hidden Markov Models (HMM) [17–19] or, more recently, deep learning [20,21]. Furthermore, apart from using multiple techniques, many different scenarios are being considered, single action recognition [22,23], group tracking [24,25], etc.

However, despite remarkable progress, the advances achieved so far do not meet high accuracy standards and the correct realization of this task in some areas, such as video surveillance, is still an open research issue.

In the analysis of a video content, many different functionalities can be implemented. One of the simplest ways to detect motion regarding a fixed background is Video Motion Detection [26–28]. Video tracking [29,30] is more challenging than the previous approach and can be very time consuming, due to the amount of data that a video contains. The aim of video tracking is to associate target objects in consecutive video frames, which can be especially difficult if the objects are moving fast in relation to

the frame rate. If object recognition techniques are needed (a challenging problem in its own), further complexity is added. On the contrary, the human brain seems to have the ability to recognize human actions perfectly. This aptitude is not just related to acquired knowledge, but also to logical reasoning and the capability of extracting relevant information from context. Based on this, the integration of commonsense reasoning [31,32] and contextual knowledge [33] has been proposed.

Hence, action recognition involves the classification of different actions from videos, a sequence of frames, taking into account as well the fact that the action could not be performed during the entire video. Although it seems an extension of image classification tasks, as it has been mentioned before, the progress for video classification has been slower due to various reasons:

- Apart from spatial information, temporal context across frames is also required.
- Huge computational cost.
- Datasets are more limited, due to the difficulty to collect, annotate and store videos.

Throughout this paper, several techniques applied for video activity recognition are mentioned, as well as the latest contributions made in the field. In addition, as a final note, some of the databases used for this topic are presented along with the results of the latest contributions using them. In Figure 1, a diagram showing the techniques explained and other tasks related to this subject but which are not discussed in this review are indicated.

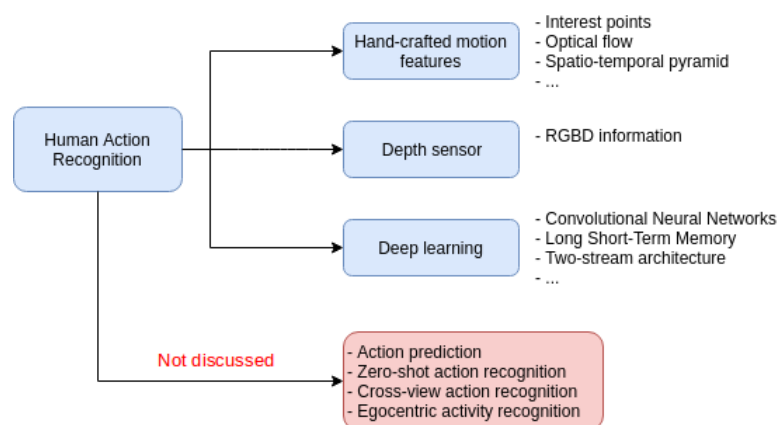


Figure 1. Summary diagram.

This review focuses on a specific area of Human Action Recognition, to keep the discussion simple. Only action recognition from a whole video recorded from a fixed position is considered in this paper, as we think this problem setup is the entrance gate to the analysis of other more complex situations, as those presented in the bottom part of Figure 1. At the same time, the complexity level of the problem considered in this review is high enough to deserve a dedicated survey. For the sake of completeness, we will briefly review the main characteristics of the situations shown in Figure 1 but not covered here. In action prediction, instead of recognizing the action that is happening in the video, the objective is to guess the action that will occur in an incomplete video. The zero-shot action recognition problem consists of training a model to classify videos of categories that have no instances in the training set, which means that there are no instances of certain classes that are going to appear in the test set. To address this issue, complementary information of invisible classes is assumed in the form of attribute vectors that describe each class. In the cross-view action recognition, there are different points of view in the scene when the action is occurring. There are other variations such as egocentric activity recognition that consists of recognizing actions from egocentric videos [34].

The survey is centered in action recognition methods for videos that are recorded in third person and the whole action occurs inside the video. Although different information can be extracted from the videos and there are articles mentioned that also use extra information such as depth sensors' information, all the presented methods have these two characteristics in common. The methods that

are explained use databases with the characteristics of the ones presented in Section 3. Although there are previous reviews on video action recognition [12,35,36], as it is a subject that is continuously progressing, it is always necessary to have a survey that collects the latest contributions. Our review, apart from mentioning articles that others have not been able to collect since they have been published later, also deals with older articles that have served as reference for later methods.

## 2. Used Techniques

As activity recognition has been an active research area lately, there have been many different approaches to deal with this problem. Throughout the survey, some of these are introduced, starting with simpler approaches and finishing with the newest contributions to the field. The proposed methods that try to solve this problem that are referred to in this paper could be separated into three main groups: methods using hand-crafted motion features, depth information based methods and deep learning based methods. Strictly speaking, these three areas are somehow interrelated and depth sensors features could lie under the hand-crafted or deep learning categorization. For a long time, computer vision has focused on data recorded from RGB (visible light) cameras, especially in the case of videos. Depth sensors have started to be used in the field of video analysis in more recent times and this is the reason why we feel it deserves a separate section.

First, hand-crafted motion features methods are explained. In these methods, some interesting features are obtained from the raw pixels of the video frames and then these features are used to perform the recognition. Second, depth information based methods are analyzed, which use depth maps as extra information. Third, deep learning methods are presented, which, unlike hand-crafted methods, achieve the features for the recognition automatically. Throughout the document, several methods that combine some of these three modalities are also presented.

### 2.1. Methods Using Hand-Crafted Motion Features

This document focuses on video-based activity recognition, in which the representation of visual and temporal information becomes important. There are several ways to extract visual features, both static image features and temporal visual features, and then use them to perform the recognition. Temporal visual features are a combination of static image features and time information, so, through these features, temporal video information is achieved. Key-frame [37,38], bag-of-words (BoW) [39,40], interest points [41,42] and motion based approaches [43–46] are types of representations that can be obtained from a video. *Key-frame* based approaches, as the name indicates, consist of detecting the key-frames of the video which would be used for classification; *BoW* based approaches represent the frames of the video segments over a vocabulary of visual features; *interest points* based approaches focus on simply selecting a specific set of points or pixels for the classification and, to finish, *motion* based approaches focus on the movement along the video. Throughout this section, only motion based approaches are analyzed.

In [47], the authors use a temporal template as the basis of their representation, continuing with their approach presented in [48]. This temporal template consists of a static vector-image where the value of the vector at each point represents a function of the motion properties at the corresponding spatial location in an image sequence. They explore their representation with a simple two component version of the template:

- The first value indicates the presence of motion and where it occurs by a binary motion-energy image (MEI). Being  $D(x, y, t)$  a binary image sequence and  $r$  the value that defines the temporal extent of a movement, the binary image is defined this way:

$$E_r(x, y, t) = \bigcup_{i=0}^{r-1} D(x, y, t - i). \quad (1)$$

- The second value is a scalar-valued image where intensity is a function of recency of motion of the sequence, represented by a motion-history image (MHI) which indicates how the image is moving.  $H_r$  represents the temporal history of motion at each point, where recently moved pixels are brighter:

$$H_r(x, y, t) = \begin{cases} r, & \text{if } D(x, y, t) = 1, \\ \max(0, H_r(x, y, t-1) - 1), & \text{otherwise.} \end{cases} \quad (2)$$

Then, a recognition method is developed, which matches these temporal templates against stored instances of known actions. They also present a recognition method to automatically perform temporal segmentation being invariant to linear changes in speed.

The authors of [49] demonstrate that local measurements in terms of spatio-temporal interest points (local features) can be used to recognize complex motion patterns. As these features, which capture local motion events in videos, can be adapted to size, frequency and velocity of moving patterns, the resulting video representations are stable with respect to the corresponding transformations. To represent motion patterns, they use local space-time features [50] and to detect local features they construct, using Gaussian convolution, its scale-space representation. Then, they explore the integration of local space-time features with Support Vector Machines (SVM) classifier [51,52], used in many visual pattern recognition methods [53,54], and apply the resulting approach to the recognition of human actions. In addition, for the purpose of evaluation, the authors introduce a new video database containing 2391 clips of six human-actions performed by 25 people in four scenarios.

In [55], the authors present a hybrid hierarchical model, inspired by [56], where video sequences are represented as collections of spatial and spatio-temporal features. These features are achieved by extracting both static and dynamic interest points and the model is able to combine static and motion image features, as well as performing categorization of human actions in a frame-by-frame basis. Motion features are extracted as in [40]. They show that using static and dynamic features together is better than using just a single feature type.

Laptev et al. [42] contribute to the recognition of realistic videos and use movie scripts for automatic annotation of human actions in videos. Due to the achievements in image classification [57–60], they employ spatio-temporal features and spatio-temporal pyramids, extending spatial pyramids of [58]. Interests points are detected as in [50] using a space-time extension of the Harris operator [61]. Then, a multi-scale approach is used and features at multiple levels are extracted. For classification, they use a nonlinear SVM with a multi-channel Gaussian kernel [60]. Apart from the action recognition task, their main contribution consists of automatically annotating human actions with the use of movie scripts and getting videos with more realistic characteristics.

Visual features such as edges, corners, interest points, etc. can be used to form a more complicated feature called optical flow. The optical flow methods try to calculate the motion between two image frames which are taken at times  $t$  and  $t + \Delta t$  at every position, assuming that the intensity of objects does not change during the movement  $I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$ . Expanding that equation using the Taylor Series Expansion [62] and further calculations, this equation is obtained:

$$I_x V_x + I_y V_y = -I_t \quad (3)$$

or

$$\nabla I^T \cdot \vec{V} = -I_t. \quad (4)$$

The solution, the optical flow, is the value of  $\vec{V}$ . Some approaches are given in the calculation of optical flow due to the fact that there are two unknowns in the equation. In this part, several methods that have made use of this feature and its variations are presented.

The authors of [63] present a method to recognize human actions observing them from a far field of view, but they also test their model with normal resolution datasets, such as Weizmann [64]. They use Histograms of Oriented Gradients (HOG) for human pose representations, first introduced

in [65] and successfully applied in multiple action recognition methods [66–69]. They also use a time series of Histogram of Oriented Optical Flow (HOOF) to characterize human motion. To get a subset of discriminantly informative principal components (PCs), an extension of Supervised Principal Component Analysis (SPCA) [70] technique is used, which tries to select a subset of PCs in order to best separate samples projected from different classes. This step significantly speeds up the run-time of recognition without sacrificing accuracy. A multi-class Support Vector Machine (SVM) classifier is trained for action classification. The classifier prediction is made by a collection of one-against-one SVM classifiers, as in the implementation of [71].

In [46], inspired by the success of histograms of features in object recognition, the authors propose the representation of each frame with the use of HOOF features, which are independent from the scale of the moving person and to the direction of motion. These histograms are created by computing optical flow at every frame and binning the vectors according to each primary angle. To classify HOOF time-series, they posit a generalization of the Binet–Cauchy kernels [72] to nonlinear dynamical systems (NLDS), as the data that represents, for instance, that the histogram time series is non-Euclidean and needs to be modeled with nonlinear dynamical systems. The generalization is done by using a Mercer kernel [73] on the output space. The Binet–Cauchy kernels are used for NDLS to perform the activity recognition and proposed HOOF features as outputs of NLDS.

The authors of [45] introduce a motion descriptor based on direction of optical flow. In their method, interest silhouettes are subtracted from the background (used dataset provides foreground masks [64]) and optical flow is computed using the Lucas–Kanade algorithm [74]. Then, before computing a direction histogram, the window is divided into eight regions. To represent the distribution of optical flow direction, they use a histogram, segmenting the direction of optical flow into eight bins. To create the motion vector, they concatenate a direction histogram of optical flow in every region. They also smooth the motion vectors to reduce motion variation and noise, and then these vectors are used for classification. K-means clustering [75] is first used to group similar postures and then the classification is done by a K-NN classifier. Niebles et al. [55] also used clustering but with a bag-of-words model instead of motion.

Due to the demonstration of dense trajectories being efficient video representations, in [76], their performance is improved by using camera motion to correct them. The estimation of camera motion is done by matching feature points between frames using SURF (speeded up robust features) descriptors [77] and dense optical flow [78]. A human detector [79,80] is used to remove inconsistent matches generated because of the differences of human and camera motions and, in addition, background trajectories are also removed. Motion-based descriptors, such as HOF (histogram optical flow), are significantly improved by this.

In [81], the authors propose a generic temporal video segment representation method for action recognition based on optical flow concept [62], with the idea that, to deal with a video-based action recognition problem, temporally represented video information is needed. In their approach, for feature detection, the Shi–Tomasi algorithm is used [82], which is based on Harris corner detector [61], and, to estimate optical flow, the Lucas–Kanade algorithm [74] is computed. For each selected frame of the video, optical flow vectors are grouped according to their angular features. Being an optical flow histogram the most common method of optical flow based video representation, they enrich these approaches by a novel velocity concept, Weighted Frame Velocity. This concept refers to the velocity of cumulative angular grouping of a temporal video segment, which represents the motion of the frames more descriptively. Similarities in the histogram do not always mean that there are similarities in the motion, so, instead of using a histogram based approach as in [46,83–85], vectors are grouped with respect to their angular characteristics and then summed and integrated with the new velocity concept.

The authors of [15] propose a local descriptor built by optical flow vectors along the edges of the action performers. First, a foreground extraction is done by a Gaussian Mixture Model (GMM) based method [86] and optical flow based technique [62] in order to segment the region of interest. To represent the segmented objects, optical flow based feature vectors are computed along the boundary

using Horn and Schunck algorithm [62] based optical flow extraction technique. This way, shape and instantaneous velocity information extracted from the boundaries of the action performers are incorporated in the feature set. These features are then used to feed a multi-class SVM classifier.

In [87], human activities are recognized using background subtraction, HOG features and Back-Propagation Neural Network (BPNN) classifier. In this approach, background estimation is performed at first, using mean filter to obtain the background and areas of the image containing important information. Afterwards, in order to extract features to describe human motion, a histogram of oriented gradients (HOG) [65] descriptor is used, with the idea that local shape information can be completely described by intensity gradients or edge directions. Finally, a BPNN is used to perform the final classification.

In Table 1, a summary of the explained methods using hand-crafted motion features is presented.

**Table 1.** Summary of methods using hand-crafted motion features.

	YEAR	SUMMARY	DATASET
Bobick et al. [47]	2001	Use of motion-energy image (MEI) and motion-history image (MHI).	-
Schuldt et al. [49]	2004	Use of local space-time features to recognize complex motion patterns.	KTH Action [49]
Niebles et al. [55]	2007	Use of a hybrid hierarchical model, combining static and dynamic features.	Weizmann [64]
Laptev et al. [42]	2008	Use of spatio-temporal features and extend spatial pyramids to spatio-temporal pyramids.	KTH Action [49] Hollywood [42]
Chen et al. [63]	2009	Use of HOG for human pose representations and HOOF to characterize human motion.	Weizmann [64] Soccer [83] Tower [63]
Chaudhry et al. [46]	2009	Use of HOOF features by computing optical flow at every frame and binning them according to primary angles.	Weizmann [64]
Lertniphonphan et al. [45]	2011	Use of a motion descriptor based on direction of optical flow.	Weizmann [64]
Wang et al. [76]	2013	Use of camera motion to correct dense trajectories.	HMDB51 [88] UCF101 [89] Hollywood2 [90] Olympic Sports [91]
Akpinar et al. [81]	2014	Use of a generic temporal video segment representation, introducing a new velocity concept: Weighted Frame Velocity.	Weizmann [64] Hollywood [42]
Kumar et al. [15]	2016	Use of a local descriptor built by optical flow vectors along the edges of the action performers.	Weizmann [64] KTH Action [49]
Sehgal, S. [87]	2018	Use of background subtraction, HOG features and BPNN classifier.	Weizmann [64]

## 2.2. Depth Information Based Methods

The interest of applying depth data captured from depth cameras for the action recognition problem has grown due to the advances of imaging technology in capturing depth information in real time, such as Microsoft Kinect [92] and Intel Realsense [93]. In the past few decades, research of human action recognition has mainly concentrated on video sequences captured by traditional RGB cameras, but, thanks to the advances in imaging techniques, RGBD sensors are able to capture color image sequences together with depth maps in real time. Depth images are insensitive to changes in lighting conditions and provide additional body shape and motion information that can help with

distinguishing actions that generate similar projections from a single view. In this paper, some of the recent methods using depth maps are introduced. However, if more information is required, there are many other interesting methods to analyze [94–97].

In [98], the authors propose the use of sequences of depth maps for action recognition, which provide additional body shape and motion information. In their approach, in order to make use of the additional body shape and motion information from depth maps, they generate Depth Motion Maps (DMM) by projecting depth maps into three orthogonal Cartesian planes and accumulating global activities through entire video sequences. Then, a good characterization of the local appearance and shape on DMM is achieved with HOG, Histogram of Oriented Gradients. HOG descriptors extracted from depth motion map of each projection view (front, top, side) are combined as DMM-HOG, which is used to represent the entire action video sequences. This DMM-HOG descriptor is the input to a linear SVM classifier which is used to make the recognition.

A new descriptor for activity recognition from videos obtained with a depth sensor is presented in [99], called the histogram of oriented 4D surface normals (HON4D). In order to capture the complex joint shape-motion cues at the pixel level, the authors use a histogram to describe depth sequence, which captures the distribution of the surface normal orientation in 4D space of time, depth and spatial coordinates. Instead of concatenating features [100], their histogram, as it operates in 4D space, captures the distribution of the changing shape and motion cues along with their correlation. The histogram is built by creating 4D projectors that represent the possible directions of the 4D normal and, as the descriptor is a representation for the entire sequence, it is robust against noise and occlusion, unlike other methods [101]. To quantize the 4D space, they use the vertices of a polychoron to get a more discriminative quantification.

In [102], the authors present a two-layer Bag-of-Visual-Words (BoVW) model. First, they delete background clutter, so background noise is removed. In addition, foreground noise disturbances are eliminated by jointly using motion and shape information. To distinguish similar actions, motion-based STIPs (spatial-temporal interest points) and shape based STIPs are detected. They use 3DLSK, first mentioned in [103], to describe local structures of motion-based STIPs, and, in order to fit better to depth data and its lack of texture or scale changes effects, they propose a multi-scale 3DLSK (M3DLSK). On the other hand, to capture spatial-temporal relationships among STIPs, they extract a spatial-temporal vector (STV) descriptor for each STIP to distinguish between different actions. Fusing both descriptors, M3DLSK and STV, a feature representation able to capture local and global motion and shape is achieved.

Satyamurthi et al. [104] propose the use of depth motion maps projected on multiple directions, multi-directional projected depth motion map (MPDMM), based on depth motion maps [96,98]. The proposed approach can be separated in three key components. First, they propose to extract features by converting the video sequences into frames using multi-directional projected DMM. The input 3D depth action video is projected into a set of 2D maps according to a set of planes and directions. After calculating the motion energy of each projected map, this is concatenated through entire video sequences to get the MPDMM model. Second, features are extracted from MPDMM model, on the basis of conventional texture-based Local Binary Patterns (LBP) descriptors [105]. The MPDMM image is processed with the LBP technique by thresholding the neighborhood of each pixel and outputting the result as a series of binary numbers that are then used as a statistical measure forming a histogram. Third, the kernel-based Extreme Learning Machine (ELM) [106] with a radial basis function kernel is applied to perform the classification.

In Table 2, a summary of the explained depth information based methods is presented.

**Table 2.** Summary of depth information based methods.

	YEAR	SUMMARY	DATASET
Yang et al. [98]	2012	Use of Depth Motion Maps (DMM), combining them with HOG descriptors.	MSRAction3D [107]
Oreifej et al. [99]	2013	Use of histogram of oriented 4D surface normals (HON4D) descriptor.	MSRAction3D [107] MSRGesture3D [108] 3D Action Pairs [99]
Liu et al. [102]	2018	Use of a two-layer BoVW model, using motion-based and shape-based STIPs to distinguish the action.	MSRAction3D [107] UTKinect-Action [109] MSRGesture3D [108] MSRDailyActivity3D [100]
Satyamurthi et al. [104]	2018	Use of multi-directional projected depth motion maps (MPDMM).	MSRAction3D [107] MSRGesture3D [108]

### 2.3. Deep Learning Based Methods

After being a breakthrough in image classification, it was a matter of time to start using deep learning for video-based activity recognition. Although great advances have been made and state-of-the-art results have been achieved, the level of image classification has not been reached yet.

In 2014, a paper was released [110] encouraged by the results of *Convolutional Neural Networks (CNNs)* [111] for image recognition problems [112–115]. Using a 1M videos dataset, they studied different ways for extending the connectivity of a CNN in a time domain in order to take advantage of local spatio-temporal information. They proposed three connectivity patterns: Early Fusion, Late Fusion and Slow Fusion. The Early Fusion extension combines information across an entire time window immediately on the pixel level. The Late Fusion model places two separate single-frame networks with shared parameters a distance of 15 frames apart and then merges the two streams in the first fully connected layer. This way motion can not be detected until the fully connected layer, which compares both outputs to compute global motion. The Slow Fusion model slowly fuses temporal information throughout the network such that higher layers get access to progressively more global information in both spatial and temporal dimensions. For optimization, Downpour Stochastic Gradient Descent [116] is used. The results show that a slow fusion model performs better than the early and late fusion alternatives. They also find out that a single-frame model already displays very strong performance, suggesting that local motion may not be critically important.

In the same year as the previous paper, another work was published [117] that has been the reference of later publications. Simonyan et al. propose a two-stream Convolutional Neural Network architecture that incorporates spatial and temporal networks. Videos can naturally be decomposed into spatial and temporal components. The spatial part provides information about scenes and objects of the video, taking as input a single frame. Nevertheless, the temporal part, which consists of stacked optical flow vectors, shows the movement of the observer (the camera) and the objects in the form of motion across the frames. This way, the authors divide the architecture into two streams. Each stream is implemented using a deep ConvNet [118]; softmax scores are combined by late fusion using a SVM [119] or averaging. It seems that training a temporal network with optical flow improves the training of just stacked frames as in [110]. However, compared to the shallow representation of [76], there are some things to improve yet.

After these two publications, and taking them as a starting point, deep learning has continued to be used for activity recognition, mainly with Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) [120].

In [121], the authors investigate if recurrent models are effective for tasks involving sequences. They propose a Long-term Recurrent Convolutional Network (LRCN) and demonstrate the value of these models for activity recognition. The LSTM unit they use is as the one described in [122]. Compared to previous models, recurrent convolutional models learn compositional representations in



space and time and not just assume a fixed visual representation or perform simple temporal averaging for sequential processing. As input, both RGB and optical flow are used and it is observed that the best results are achieved by the weighted scores of both inputs as in [117]. They show that learning sequential dynamics with a deep sequence model improves previous methods that only took into account parameters of the visual domain.

Wang et al. in their work [123] presented very deep two-stream ConvNets in order to improve the results of recent architectures [117] getting closer to image domain deep models. Apart from using two known architectures, GoogLeNet [124] and VGGNet-16 [125], they use 10-frame stacking of optical flow for the temporal network and a single frame image for the spatial network. As the training datasets are small, the model is initialized by pre-training it with ImageNet and, to avoid over-fitting, dropout and data augmentation techniques are used. They proposed two new data augmentation methods: one of them consists of cropping four corners and one center of the images and, in the other, multi-scale cropping is used.

In [126], trajectory-pooled deep-convolutional descriptor (TDD) is introduced, which combines the works of [76,117]. The authors first train two-stream ConvNets and use them as feature extractors to achieve convolutional spatial and temporal feature maps from the learned networks. With the improved trajectories method, a set of point trajectories are detected and, using trajectory pooling, TDD descriptors are created based on normalized convolutional feature maps and these trajectories, as in Equation (5):

$$D(T_k, \tilde{C}_m^a) = \sum_{p=1}^P \tilde{C}_m^a(\overline{(r_m \times x_p^k)}, \overline{(r_m \times y_p^k)}, z_p^k), \quad (5)$$

where  $T_k$  is a trajectory,  $\tilde{C}_m^a$  is a  $m$ th layer normalized feature map,  $(x_p^k, y_p^k, z_p^k)$  is the  $p$ th point position of video coordinates of  $T_k$  trajectory and  $r_m$  is the  $m$ th layer map size ratio,  $(\overline{\cdot})$  being the rounding operation. Fisher vector representation is used to bring together TDDs over the whole video and, finally, an SVM classifier does the recognition.

Although having some similarities with previous works [110,117], in [127], Tran et al., instead of using 2D convolutions across frames, use 3D convolutions and 3D pooling, propagating temporal information across all the layers in the network. They propose a simple yet effective approach for spatio-temporal feature learning using deep three-dimensional, convolutional networks trained on a large scale supervised video dataset. They show that 3D ConvNets [110,128] with a linear classifier are more suitable for spatio-temporal feature learning than 2D ConvNets and that the model performs even better additionally using hand-crafted features like iDT [76].

In the work by Feichtenhofer et al. [129], the authors add two ideas to the two-stream architecture of [117]. They show that it is important to associate spatial feature maps of a particular area to temporal feature maps for that corresponding region. The spatial and temporal networks are fused at an early level, so, rather than fusing at the softmax layer, they are fused at a convolutional layer. The fusion can be made in different ways and, in [130], Yue-Hei et al. evaluate many other methods to combine two-stream ConvNets over time. The architecture they propose does not increase the number of parameters significantly compared to previous methods and their results are improved by adding also iDT features [76].

Wang et al. also improved the two streams architecture in their work [131], presenting a long-rate temporal structure model, the Temporal Segment Network (TSN). Most of the previous works were not able to incorporate long-range temporal structures, but their model combines a sparse temporal sampling strategy and video-level supervision to enable efficient and effective learning using the whole action video. Another problem they wanted to deal with was over-fitting because, due to the difficulty of collecting data, the available datasets were limited. They use different techniques to avoid the risk of over-fitting: batch normalization [132], dropout [133] and pre-training. The authors also evaluate the model using four different input modalities: optical-flow, warped optical-flow, RGB and RGB difference, the last one inspired by [134].

Bilen et al. [135] presented the concept of dynamic image, which summarizes a video into just a single RGB image by applying rank pooling on the images of a video. This way, image classification CNNs can be used directly, as the input is an image. The idea of reducing the whole video to a single image is taken from [136]. In their experiments, two scenarios were considered: getting a single dynamic figure from a video or getting several dynamic images from each video, the second approach is thought to deal with the lack of training videos. Then, dynamic feature maps are obtained by adding a new temporal layer to the CNN and a pre-trained CaffeNet [137] model is used to initialize the network.

In 2017, Carreira et al. [138] presented a new architecture that uses two different 3D networks for both streams of a two-stream architecture [117], called Two-Stream Inflated 3D ConvNet (I3D). It is based on 2D ConvNet inflation, expanding filters and pooling kernels of very deep image classification ConvNets into 3D, leading to very deep spatio-temporal classifiers and making it possible to learn spatio-temporal feature extractors from videos. In basic two-stream architectures the spatial stream is formed by single frames; however, in I3D, the spatial stream input consists of frames stacked in time dimension. Apart from the new model, the main contribution of this paper is a new dataset for action recognition, the Kinetics Human Action Video dataset, which is two orders of magnitude larger than previous datasets with 400 actions and more than 400 clips per action collected from YouTube. They also showed that, when pre-training on Kinetics, results of I3D models are improved.

Later in 2018, [139] improved the performance of [121] by using lower spatial resolution and longer clips to keep the complexity of networks tractable while dealing with the inability to capture long range temporal information. They consider space-time convolutional neural networks [127,128,140] and study architectures with long-term temporal convolutions (LTC), which are used to learn video representations. As in [121], different low-level representations are studied: RGB and optical flow. Their experiments confirm the advantage of motion-based representations and highlight the importance of good quality motion estimation for learning efficient representations for human action recognition.

Ullah et al. [141] proposed an action recognition method by processing the video data using convolutional neural networks (CNN) and deep bidirectional LSTM (DB-LSTM) networks [142]. On the one hand, in order to reduce complexity and redundancy, deep features are extracted from every six frames of a video using pre-trained AlexNet [112]. Then, the sequential information among frame features is learned using an DB-LSTM network, where multiple layers are stacked together in both forward pass and backward pass of DB-LSTM to increase its depth. The video is analyzed in  $N$  chunks and  $N$  depends on processing time interval  $T$ . The final output is the combination of small chunks outputs. As the video is processed and features are analyzed for a certain time interval, the proposed method is able to learn long sequences and recognize actions in long videos.

Wang et al. [143] proposed a discriminative pooling based on the idea that, among the frames, not all of them have the same importance and a few are those that provide characteristic information about the action [144]; some of the features in one sequence are indeed useful, while the rest are not. Taking all the CNN features as positive (containing good and bad features) and the known background or noisy frames as negative, a nonlinear hyperplane that differentiates the discriminative features from the rest is learned to make the separation. The decision boundary of the classifier thus learned is then used as a descriptor for the entire video sequence, which they call the SVM Pooled (SVMP) descriptor. Thus, they formulate an efficient solver that learns these hyperplanes per video and the corresponding action classifiers over the hyperplanes. This pooling scheme is end-to-end trainable within a deep framework.

The authors of [145] presented the first end-to-end convNets which admit videos of arbitrary size and length. After seeing that 3D convolutional networks have achieved good results in action recognition, they decided to delete two of the requirements that existing convNets had: fixed size and length input videos were required, which reduce the quality of video analysis. Basically, each video is decomposed into spatial and temporal shots and, for both pieces of information, the same process is computed. A spatial temporal pyramid pooling (STPP) convNet is first used to extract

equal-dimensional descriptors from variable-sized frame sequences. Then, a Long Short-Term Memory (LSTM) or a CNN-E model is used to recognize the actions from these descriptors. Finally, both streams (spatial and temporal) are combined by a late fusion.

In Table 3, a summary of the deep learning based methods explained is presented.

**Table 3.** Summary of deep learning based methods.

	YEAR	SUMMARY	DATASET
Karpathy et al. [110]	2014	Use of different connectivity patterns for CNNs: early fusion, late fusion and slow fusion.	Sports-1M [110] UCF101 [89]
Simonyan et al. [117]	2014	Use of a two-stream CNN architecture, incorporating spatial and temporal networks.	UCF101 [89] HMDB51 [88]
Donahue et al. [121]	2015	Use of a Long-term Recurrent Convolutional Network (LRCN) to learn compositional representations in space and time.	UCF101 [89]
Wang et al. [123]	2015	Use of very deep two-stream convNets, using stacked optical flow for temporal network and a single frame image for spatial network.	UCF101 [89]
Wang et al. [126]	2015	Use of trajectory-pooled deep-convolutional descriptor (TDD).	UCF101 [89] HMDB51 [88]
Tran et al. [127]	2015	Use of deep 3D convolutional networks, which are better for spatio-temporal feature learning.	UCF101 [89]
Feichtenhofer et al. [129]	2016	Use of two-stream architecture associating spatial feature maps of a particular area to temporal feature maps of that region and fusing the networks at an early level.	UCF101 [89] HMDB51 [88]
Wang et al. [131]	2016	Use of Temporal Segment Network (TSN) to incorporate long-range temporal structures avoiding overfitting.	UCF101 [89] HMDB51 [88]
Bilen et al. [135]	2016	Use of image classification CNNs after summarizing the videos in dynamic images.	UCF101 [89] HMDB51 [88]
Carreira et al. [138]	2017	Use of two-stream Inflated 3D ConvNet (I3D), using two different 3D networks for both streams of a two-stream architecture.	UCF101 [89] HMDB51 [88]
Varol et al. [139]	2018	Use of space-time CNNs and architectures with long-term temporal convolutions (LTC), using lower spatial resolution and longer clips.	UCF101 [89] HMDB51 [88]
Ullah et al. [141]	2018	Use of CNNs to reduce complexity and redundancy and deep bidirectional LSTM (DB-LSTM) to learn sequential information among frame features.	UCF101 [89] HMDB51 [88] YouTube actions [146]
Wang et al. [143]	2018	Use of a discriminative pooling, taking into account that just a few frames provide characteristic information about the action.	HMDB51 [88]
Wang et al. [145]	2018	Use of convNets which admit videos of arbitrary size and length, using first a STPP and a LSTM (or CNN-E) then.	UCF101 [89] HMDB51 [88] ACT [147]

Finally, in order to compare the presented techniques briefly, some advantages and disadvantages are presented in Table 4.

**Table 4.** Advantages and disadvantages of presented techniques.

	Advantages	Disadvantages
Hand-crafted motion features	<ul style="list-style-type: none"> <li>- There is no need of a large amount of data for training.</li> <li>- It is simple and unambiguous to understand the model and analyze and visualize the functions.</li> <li>- The features used to train the model are explicitly known.</li> </ul>	<ul style="list-style-type: none"> <li>- Usually these features are not robust.</li> <li>- They can be computationally intensive due to the high dimensions.</li> <li>- The discriminative power is usually low.</li> </ul>
Depth information	<ul style="list-style-type: none"> <li>- The 3D structure information of the image that depth sensors provide is used to recover postures and recognize the activity.</li> <li>- The skeletons extracted from depth maps are precise.</li> <li>- Depth sensors can work in darkness.</li> </ul>	<ul style="list-style-type: none"> <li>- Depth maps have no texture, making it difficult to apply local differential operators.</li> <li>- The global features can be unsettled because depth maps may contain occlusions.</li> </ul>
Deep Learning	<ul style="list-style-type: none"> <li>- There is no need of expert knowledge to get suitable features, reducing the effort of feature extraction.</li> <li>- Instead of designing them manually, features are automatically learned through the network.</li> <li>- Deep neural networks can extract high-level representation in deep layer, making it more suitable for complex tasks.</li> </ul>	<ul style="list-style-type: none"> <li>- Need to collect massive data, consequently there is a lack of data sets.</li> <li>- Time consuming.</li> <li>- Problem of models capability of generalization.</li> </ul>

### 3. Benchmark Datasets

Although there is not a standard benchmark in activity recognition, there are some datasets that are being considered as reference ones [148]. As it has been mentioned above, due to the complexity of collecting data, the available datasets are limited. In this section, the most used datasets are presented.

#### 3.1. UCF-101

UCF101 [89,149] is an action recognition dataset of realistic action videos. It is composed of 13,320 videos with 101 action categories and 27 h of video data. This dataset is an extension of the UCF50 [150] dataset that has 50 action categories.

The videos have been collected from YouTube, making the dataset realistic, and it provides a great variety of videos with different objects, camera motion, background, lighting, viewpoint, etc. Based on those features, videos are gathered into 25 groups (4–7 videos per action in each group) with videos sharing some of the features, as background, for example.

The 101 categories can be divided in five main groups:

1. Human–Object Interaction: twenty categories.
2. Body-Motion Only: sixteen categories.
3. Human–Human Interaction: five categories.
4. Playing Musical Instruments: ten categories.
5. Sports: fifty categories.

#### 3.2. HMDB51

HMDB51 [88,151] is another action recognition database that collects videos from various sources, mainly from movies but also from public databases such as YouTube, Google and Prelinger Archives.

It consists of 6849 videos with 51 action categories and a minimum of 101 clips belong to each category. The action categories can be divided as well in five main groups:

1. General facial actions: smile, laugh, chew, talk.
2. Facial actions with object manipulation: smoke, eat, drink.
3. General body movements: cartwheel, clap hands, climb, climb stairs, dive, fall on the floor, backhand flip, handstand, jump, pull up, push up, run, sit down, sit up, somersault, stand up, turn, walk, wave.
4. Body movements with object interaction: brush hair, catch, draw sword, dribble, golf, hit something, kick ball, pick, pour, push something, ride bike, ride horse, shoot ball, shoot bow, shoot gun, swing baseball bat, sword exercise, throw.
5. Body movements for human interaction: fencing, hug, kick someone, kiss, punch, shake hands, sword fight.

Apart from the action label, other meta-labels are indicated in each clip. These labels provide information about some features describing properties of the clip, such as camera motion, lighting conditions, or background. As videos are taken from movies or YouTube, the variation of features is high and that extra information can be useful. In addition, the quality of the videos has been measured (*good*, *medium*, *bad*), and they are rated depending on whether body parts vanish while action is executed or not.

### 3.3. Weizmann

Before the two previous databases were created, many methods used the Weizmann [152] database published by [64] to evaluate the performance of their contributions. It provides 90 low-resolution ( $180 \times 144$ , deinterlaced 50 fps) video sequences. These clips show 10 different actions performed by nine different people. These are the actions that appear in the database: *run*, *walk*, *skip*, *jumping-jack (jack)*, *jump-forward-on-two-kegs (jump)*, *jump-in-place-on-two-legs (pjump)*, *side-gallop (side)*, *wave-two-hands (wave2)*, *wave-one-hand (wave1)* and *bend*. Background and the viewpoint are statics.

### 3.4. MSRAction3D

In 2010, as there was no public benchmark database, the authors in [107] published the database called MSRAction3D [153] which provided the sequences of depth maps captured by a depth camera. The dataset contains twenty actions: *high arm wave*, *horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *side-boxing*, *bend*, *forward kick*, *side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing*, *pick up* and *throw*. Seven different individuals performed each action three times, facing the camera during the performance. The depth maps have a size of  $640 \times 480$  and they were captured at about 15 frames per second (fps) by a depth camera with infra-red light structure.

### 3.5. ActivityNet

The authors of [154] presented in 2015 the ActivityNet [155] database. It is composed of 203 different classes with an average of 137 videos per class and a total of 648 video hours. The videos were obtained from online video sharing sites and they are around 5–10 min long. Half of the videos are in HD resolution ( $1280 \times 720$ ) and most of them have a frame rate of 30 fps.

The aim of this database is to collect activities of humans daily life and it has a hierarchical structure, organizing the activities according to social interactions and where they take place.

### 3.6. Something Something

Later, in 2017, the authors of [156] introduced the “Something Something” [157] dataset. The first version of the database consists of 108,499 videos belonging to 174 different labels with 23,137 distinct object names. The length of the videos variate between 2 and 6 s and they have a height of 100 px and variable width. Labels are textual descriptions such as “Putting *something* next to *something*”

where *something* refers to an object name. This database is already split into train, validation and test, containing 86,017, 11,522 and 10,960 videos, respectively.

However, there has been a second release of the dataset and now it contains 220,847 videos, 168,913 for the training set, 24,777 for the validation set and 27,157 for the test set. The number of labels remains the same, but there are additional object annotations now. Moreover, the pixel resolution has increased from 100 px to 240 px.

### 3.7. Sports-1M

In [110], Karpathy et al. presented a new database, Sports-1M [158], which contains 1,133,158 video URLs with 487 automatically annotated different labels. YouTube Topics API was used to do the annotation. There are around 1000–3000 videos per class and some of them, nearly the 5%, are labelled with more than one class.

Nowadays, the YouTube-8M [159] dataset is also available and the Sports-1M dataset is included in it. This dataset is composed of videos from 3862 labels and it contains 350,000 h of video. In this case, each video has an average of three labels.

### 3.8. AVA

The authors of [160] presented AVA [161], a video dataset of spatio-temporally localized Atomic Visual Actions. This dataset consists of 430 movie clips of 15 min length annotated with 80 actions (14 poses, 17 person–person, 49 person–object). There are 386,000 labelled segments, 614,000 labelled bounding boxes and 81,000 person tracks, with a total of 1.58M labelled actions, with multiple labels per person occurring frequently.

Every person of the scene is localized by a bounding box and labels are assigned according to the action performed by the actor. Each scene can have more than a label, one of them corresponds to the actor's pose and additional labels which correspond to person–object or person–person interactions can be assigned. A frame containing more than one actor is labelled separately for each person of the scene.

To finish, in Table 5, a summary of the explained datasets is introduced, in order to present the information more clearly.

**Table 5.** Summary of the presented datasets.

	# Classes	# Videos	# Actors	Resolution	Year
Weizmann	10	90	9	180 × 144	2005
MSRAAction3D	20	420	7	640 × 480	2010
HMDB51	51	6849	-	320 × 240	2011
UCF50	50	6676	-	-	2012
UCF101	101	13,320	-	320 × 240	2012
Sports-1M	487	1,133,158	-	-	2014
ActivityNet	203	27,801	-	1280 × 720	2015
Something Something	174	220,847	-	— (Variable width) × 240	2017
AVA	80	430	-	-	2018

## 4. Results

To better analyze the explained methods and the contributions of each one of them, the results obtained for mentioned datasets are compared. For each method, the achieved accuracy values for different datasets are shown, together with the reference to the original article where they have been proposed.

On the one hand, in Table 6, results for depth information based methods can be observed. These methods use the MSRAAction3D as benchmarks because the input they need is different from other models. Regarding the methods used with the MSRAAction3D database, the best result of the presented methods is achieved by [102], as it can be seen in Table 6.

**Table 6.** Obtained accuracies for the benchmark dataset with depth information based methods.

	METHOD	MSRAAction3D
DS	DMM-HOG [98]	85.52%
	HON4D [99]	88.89%
	M3DLSK+STV [102]	<b>95.36%</b>
	MPDMM [104]	94.8%

On the other hand, most of the hand-crafted feature methods use the Weizmann dataset as a benchmark. However, some of the presented models work with both UCF101 and HMDB51 datasets, which are used as benchmarks in deep learning methods. Thus, in Table 7, the obtained accuracy values can be observed, together for deep learning and hand-crafted methods.

**Table 7.** Obtained accuracies for the benchmark datasets with hand-crafted methods and deep learning methods.

	METHOD	UCF101	HMDB51	Weizmann
Hand-crafted	Hierarchical [55]	-	-	72.8%
	Far Field of View [63]	-	-	<b>100%</b>
	HOOF NLDS [46]	-	-	94.4%
	Direction HOF [45]	-	-	79.17%
	iDT [76]	-	57.2%	-
	iDT+FV [76]	85.9%	57.2%	-
	OF Based [81]	-	-	90.32%
	Edges OF [15]	-	-	95.69%
Deep learning	HOG features [87]	-	-	99.7%
	Slow Fusion CNN [110]	65.4%	-	-
	Two stream (avg) [117]	86.9%	58.0%	-
	Two stream (SVM) [117]	88.0%	59.4%	-
	IDT+MIFS [162]	89.1%	65.1%	-
	LRCN (RGB) [121]	68.2%	-	-
	LRCN (FLOW) [121]	77.28%	-	-
	LRCN (avg, 1/2-1/2) [121]	80.9%	-	-
	LRCN (avg, 1/3-2/3) [121]	82.34%	-	-
	Very deep two-stream (VGGNet-16) [123]	91.4%	-	-
	TDD [126]	90.3%	63.2%	-
	TDD + iDT [126]	91.5%	65.9%	-
	C3D [127]	85.2%	-	-
	C3D + iDT [127]	90.4%	-	-
	TwoStreamFusion [129]	92.5%	65.4%	-
	TwoStreamFusion+iDT [129]	93.5%	69.2%	-
	TSN (RGB+FLOW) [131]	94.0%	68.5%	-
	TSN (RGB+FLOW+WF) [131]	94.2%	69.4%	-
	Dynamic images + iDT [135]	89.1%	65.2%	-
	Two-StreamI3D [138]	93.4%	66.4%	-
	Two-StreamI3D, pre-trained [138]	<b>97.9%</b>	80.2%	-
	LTC (RGB) [139]	82.4%	-	-
	LTC (FLOW) [139]	85.2%	59.0%	-
	LTC(FLOW+RGB) [139]	91.7%	64.8%	-
	LTC(FLOW+RGB)+iDT [139]	92.7%	67.2%	-
	DB-LSTM [141]	91.21%	<b>87.64%</b>	-
	Two-Stream SVMP(VGGNet) [143]	-	66.1%	-
	Two-Stream SVMP(ResNet) [143]	-	71.0%	-
	Two-Stream SVMP(+ iDT) [143]	-	72.6%	-
	Two-Stream SVMP(I3D conf) [143]	-	83.1%	-
	STPP + CNN-E (RGB) [145]	85.6%	62.1%	-
	STPP + LSTM (RGB) [145]	85.0%	62.5%	-
STPP + CNN-E (FLOW) [145]	83.2%	55.4%	-	
STPP + LSTM (FLOW) [145]	83.8%	54.7%	-	
STPP + CNN-E (RGB+FLOW) [145]	92.4%	70.5%	-	
STPP + LSTM (RGB+FLOW) [145]	92.6%	70.3%	-	

As it can be seen in Table 7, the Two-Stream I3D method [138], pre-trained with the Kinetics dataset, provides the best result for the UCF101 dataset. For HMDB51, the best result is achieved by the DB-LSTM model [141] and, among those who have tested with the Weizmann database, the best value is given by the method presented in [63].

## 5. Discussion

After reading the previous sections, the researchers could ask themselves which are the most promising lines of research in the field of action recognition in videos, or where it is more likely to get a higher return for the invested effort.

For people just interested in applying an existing method to their data, or in minimal modifications or customizations, some authors of the presented methods have made their code available that can be used. These implementations are indicated in Table 8. Among them, the methods explained in [138,141] provide the best results.

**Table 8.** Available code for presented methods.

METHOD	YEAR	PAPER	CODE
Deep Learning	2018	Video representation learning using discriminative pooling [143]	SVMP <a href="https://github.com/3xWangDot/SVMP">https://github.com/3xWangDot/SVMP</a>
Deep Learning	2018	Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features [141]	Bi-directional LSTM <a href="https://github.com/Aminullah6264/BidirectionalLSTM">https://github.com/Aminullah6264/BidirectionalLSTM</a>
Deep Learning	2018	Long-term temporal convolutions for action recognition [139]	LTC <a href="https://github.com/gulvarol/ltc">https://github.com/gulvarol/ltc</a>
Deep Learning	2017	Quo vadis, action recognition? A new model and the Kinetics dataset [138]	Two-Stream I3D <a href="https://github.com/deepmind/kinetics-i3d">https://github.com/deepmind/kinetics-i3d</a>
Deep Learning	2016	Dynamic image networks for action recognition [135]	Dynamic images <a href="https://github.com/hbilen/dynamic-image-nets">https://github.com/hbilen/dynamic-image-nets</a>
Deep Learning	2016	Temporal segment networks: Towards good practices for deep action recognition [131]	TSN <a href="https://github.com/yjxiong/temporal-segment-networks">https://github.com/yjxiong/temporal-segment-networks</a>
Deep Learning	2016	Convolutional two-stream network fusion for video action recognition [129]	Two-Stream Fusion <a href="https://github.com/feichtenhofer/twostreamfusion">https://github.com/feichtenhofer/twostreamfusion</a>
Deep Learning	2015	Learning spatiotemporal features with 3D convolutional networks [127]	C3D <a href="https://github.com/facebook/C3D">https://github.com/facebook/C3D</a>
Deep Learning	2015	Action recognition with trajectory-pooled deep-convolutional descriptors [126]	TDD <a href="https://github.com/wanglimin/tdd/">https://github.com/wanglimin/tdd/</a>
Deep Learning	2015	Towards good practices for very deep two-stream convNets [123]	Very deep Two-Stream convNets <a href="https://github.com/yjxiong/caffe/tree/action_recog">https://github.com/yjxiong/caffe/tree/action_recog</a>
Depth information	2013	HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences [99]	HON4D <a href="http://www.cs.ucf.edu/~oreifej/HON4D.html">http://www.cs.ucf.edu/~oreifej/HON4D.html</a>
Hand-crafted motion features	2013	Action Recognition with Improved Trajectories [76]	Improved Trajectories <a href="http://lear.inrialpes.fr/~wang/improved_trajectories">http://lear.inrialpes.fr/~wang/improved_trajectories</a>



For researchers interested in developing new methods or in deep modifications of current ones, deep learning looks like the way to go, although the computations costs could be forbidding.

Some researchers could have the resources to generate new datasets similar to those presented in this paper. They need to have in mind several decisions: will the videos have the same resolution and/or length and will they be recorded with the same background? We advocate for datasets covering different kinds of videos, with that information present in the metadata and with enough samples of each type. Anyway, if the researcher resources are somewhat limited, which is usually the case, it is advisable to focus on just one type of video. All the technical information about the sensor might appear in the metadata, as well as lighting conditions or any information of interest. If depth sensors are used, high and low resolution of the depth data could be provided. Processing of depth data can be computationally expensive, and other researchers using that dataset could benefit from access to a standard low resolution version of that data.

The task of labeling the database samples can be eased with the help of some tools. While just providing a global label for a video does not require a great deal of effort, the video database curators could choose to gather information about individual frames in the videos. There are several tools that could be useful in this task, like Sloth [163], LabelMe [164,165] or LabelBox [166].

It is difficult to predict the future development of this area, but, at least in the short term, the overall tendency in machine learning is going towards massive data, computationally expensive algorithms and dedicated hardware. It is expected that the price of depth sensors will keep a descending curve, as well as the cost of hardware in general. The main challenges are expected to be twofold: for the researchers developing new methods, those related to the storage and processing of massive databases, and for developers integrating the methods into software solutions, those related to a fast classification time.

## 6. Conclusions

In this paper, different methods for video activity recognition have been presented. Several models have been explained showing the development of recent years. Likewise, several databases used to evaluate the performance of the models have been introduced. The results have been shown together in a table in order to compare the methods presented correctly.

Due to the extent width of the subject, there are many more models that have not been mentioned in this document. Even so, an attempt has been made to show a current state-of-the-art by presenting different techniques to deal with the problem. To sum up, through this document, we have tried to show the relevance and current situation of video-based activity recognition.

Video-based activity recognition, as it has been mentioned before, is more complicated than static image classification and this is also reflected in the results obtained so far. However, since deep learning is still being exploited, in the near future, this task may become easier to perform and current results may be improved using some deep learning techniques.

**Author Contributions:** B.S. and J.M.M.-O. supervised the paper. I.R.-M. collected the materials and wrote the paper. I.R. and E.J. revised and formatted the paper.

**Funding:** This work has been partially supported by the Basque Government, Spain (IT900-16), the Spanish Ministry of Economy and Competitiveness (RTI2018-093337-B-I00 , MINECO/FEDER, EU).

**Acknowledgments:** The authors thank the editor and three anonymous reviewers for their comments on the early version of this manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Avci, A.; Bosch, S.; Marin-Perianu, M.; Marin-Perianu, R.; Havinga, P. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In Proceedings of the 23th International Conference on Architecture of Computing Systems 2010, Hannover, Germany, 22–23 February 2010; pp. 1–10.
2. Mulroy, S.; Gronley, J.; Weiss, W.; Newsam, C.; Perry, J. Use of cluster analysis for gait pattern classification of patients in the early and late recovery phases following stroke. *Gait Posture* **2003**, *18*, 114–125. [[CrossRef](#)]
3. Rautaray, S.S.; Agrawal, A. Vision based hand gesture recognition for human computer interaction: A survey. *Artif. Intell. Rev.* **2015**, *43*, 1–54. [[CrossRef](#)]
4. Mitra, S.; Acharya, T. Gesture recognition: A survey. *IEEE Trans. Syst. Man Cybern. Part Appl. Rev.* **2007**, *37*, 311–324. [[CrossRef](#)]
5. Vishwakarma, S.; Agrawal, A. A survey on activity recognition and behavior understanding in video surveillance. *Vis. Comput.* **2013**, *29*, 983–1009. [[CrossRef](#)]
6. Leo, M.; D’Orazio, T.; Spagnolo, P. Human activity recognition for automatic visual surveillance of wide areas. In Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks, New York, NY, USA, 15 October 2004; pp. 124–130.
7. Coppola, C.; Cosar, S.; Faria, D.R.; Bellotto, N. Social Activity Recognition on Continuous RGB-D Video Sequences. *Int. J. Soc. Robot.* **2019**, 1–15. [[CrossRef](#)]
8. Coppola, C.; Faria, D.R.; Nunes, U.; Bellotto, N. Social activity recognition based on probabilistic merging of skeleton features with proximity priors from RGB-D data. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 5055–5061.
9. Lin, W.; Sun, M.T.; Poovandran, R.; Zhang, Z. Human activity recognition for video surveillance. In Proceedings of the 2008 IEEE International Symposium on Circuits and Systems, Seattle, WA, USA, 18–21 May 2008; pp. 2737–2740.
10. Nair, V.; Clark, J.J. Automated visual surveillance using Hidden Markov Models. *International Conference on Vision Interface*. 2002, pp. 88–93. Available online: <https://pdfs.semanticscholar.org/8fcf/7e455419fac79d65c62a3e7f39a945fa5be0.pdf> (accessed on 15 July 2019).
11. Ma, M.; Meyer, B.J.; Lin, L.; Proffitt, R.; Skubic, M. VicoVR-Based Wireless Daily Activity Recognition and Assessment System for Stroke Rehabilitation. In Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Madrid, Spain, 3–6 December 2018; pp. 1117–1121.
12. Ke, S.R.; Thuc, H.; Lee, Y.J.; Hwang, J.N.; Yoo, J.H.; Choi, K.H. A review on video-based human activity recognition. *Computers* **2013**, *2*, 88–131. [[CrossRef](#)]
13. Dawn, D.D.; Shaikh, S.H. A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector. *Vis. Comput.* **2016**, *32*, 289–306. [[CrossRef](#)]
14. Herath, S.; Harandi, M.; Porikli, F. Going deeper into action recognition: A survey. *Image Vis. Comput.* **2017**, *60*, 4–21. [[CrossRef](#)]
15. Kumar, S.S.; John, M. Human activity recognition using optical flow based feature set. In Proceedings of the 2016 IEEE International Carnahan Conference on Security Technology (ICCST), Orlando, FL, USA, 24–27 October 2016; pp. 1–5.
16. Guo, K.; Ishwar, P.; Konrad, J. Action recognition using sparse representation on covariance manifolds of optical flow. In Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, Boston, MA, USA, 29 August–1 September 2010; pp. 188–195.
17. Niu, F.; Abdel-Mottaleb, M. HMM-based segmentation and recognition of human activities from video sequences. In Proceedings of the 2005 IEEE International Conference on Multimedia and Expo, Amsterdam, The Netherlands, 6 July 2005; pp. 804–807.
18. Raman, N.; Maybank, S.J. Activity recognition using a supervised non-parametric hierarchical HMM. *Neurocomputing* **2016**, *199*, 163–177. [[CrossRef](#)]
19. Liciotti, D.; Duckett, T.; Bellotto, N.; Frontoni, E.; Zingaretti, P. HMM-based activity recognition with a ceiling RGB-D camera. In Proceedings of the ICPRAM—6th International Conference on Pattern Recognition Applications and Methods, Porto, Portugal, 24–26 February 2017.
20. Ma, M.; Fan, H.; Kitani, K.M. Going deeper into first-person activity recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1894–1903.

21. Nunez, J.C.; Cabido, R.; Pantrigo, J.J.; Montemayor, A.S.; Velez, J.F. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognit.* **2018**, *76*, 80–94. [[CrossRef](#)]
22. Sadanand, S.; Corso, J.J. Action bank: A high-level representation of activity in video. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1234–1241.
23. Ng, J.Y.H.; Davis, L.S. Temporal difference networks for video action recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1587–1596.
24. Lan, T.; Sigal, L.; Mori, G. Social roles in hierarchical models for human activity recognition. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1354–1361.
25. Vahora, S.; Chauhan, N. Deep neural network model for group activity recognition using contextual relationship. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 47–54. [[CrossRef](#)]
26. Huang, S.C. An advanced motion detection algorithm with video quality analysis for video surveillance systems. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *21*, 1–14. [[CrossRef](#)]
27. Hu, W.; Tan, T.; Wang, L.; Maybank, S. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. Syst. Man Cybern. Part Appl. Rev.* **2004**, *34*, 334–352. [[CrossRef](#)]
28. Gaba, N.; Barak, N.; Aggarwal, S. Motion detection, tracking and classification for automated Video Surveillance. In Proceedings of the 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, 4–6 July 2016; pp. 1–5.
29. Trucco, E.; Plakas, K. Video tracking: A concise survey. *IEEE J. Ocean. Eng.* **2006**, *31*, 520–529. [[CrossRef](#)]
30. Maggio, E.; Cavallaro, A. *Video Tracking: Theory and Practice*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
31. Del Rincón, J.M.; Santofimia, M.J.; Nebel, J.C. Common-sense reasoning for human action recognition. *Pattern Recognit. Lett.* **2013**, *34*, 1849–1860. [[CrossRef](#)]
32. Santofimia, M.J.; Martinez-del Rincon, J.; Nebel, J.C. Episodic reasoning for vision-based human action recognition. *Sci. World J.* **2014**, *2014*. [[CrossRef](#)]
33. Onofri, L.; Soda, P.; Pechenizkiy, M.; Iannello, G. A survey on using domain and contextual knowledge for human activity recognition in video streams. *Expert Syst. Appl.* **2016**, *63*, 97–111. [[CrossRef](#)]
34. Wang, X.; Gao, L.; Song, J.; Zhen, X.; Sebe, N.; Shen, H.T. Deep appearance and motion learning for egocentric activity recognition. *Neurocomputing* **2018**, *275*, 438–447. [[CrossRef](#)]
35. Aggarwal, J.K.; Ryoo, M.S. Human activity analysis: A review. *ACM Comput. Surv. (CSUR)* **2011**, *43*, 16. [[CrossRef](#)]
36. Kong, Y.; Fu, Y. Human Action Recognition and Prediction: A Survey. *arXiv* **2018**, arXiv:1806.11230.
37. Raptis, M.; Sigal, L. Poselet key-framing: A model for human activity recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2650–2657.
38. Wang, Y.; Sun, S.; Ding, X. A self-adaptive weighted affinity propagation clustering for key frames extraction on human action recognition. *J. Vis. Commun. Image Represent.* **2015**, *33*, 193–202. [[CrossRef](#)]
39. Niebles, J.C.; Wang, H.; Fei-Fei, L. Unsupervised learning of human action categories using spatial-temporal words. *Int. J. Comput. Vis.* **2008**, *79*, 299–318. [[CrossRef](#)]
40. Dollár, P.; Rabaud, V.; Cottrell, G.; Belongie, S. Behavior recognition via sparse spatio-temporal features. In Proceedings of the 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Beijing, China, 15–16 October 2005; pp. 65–72.
41. Bregonzio, M.; Gong, S.; Xiang, T. Recognising action as clouds of space-time interest points. In Proceedings of the CVPR 2009, Miami Beach, FL, USA, 20–25 June 2009; Volume 9, pp. 1948–1955.
42. Laptev, I.; Marszalek, M.; Schmid, C.; Rozenfeld, B. Learning realistic human actions from movies. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008, pp. 1–8.
43. Ngo, C.W.; Pong, T.C.; Zhang, H.J. Motion-based video representation for scene change detection. *Int. J. Comput. Vis.* **2002**, *50*, 127–142. [[CrossRef](#)]
44. Sand, P.; Teller, S. Particle video: Long-range motion estimation using point trajectories. *Int. J. Comput. Vis.* **2008**, *80*, 72. [[CrossRef](#)]

45. Lertniphonphan, K.; Aramvith, S.; Chalidabhongse, T.H. Human action recognition using direction histograms of optical flow. In Proceedings of the 2011 11th International Symposium on Communications & Information Technologies (ISCIT), Hangzhou, China, 12–14 October 2011; pp. 574–579.
46. Chaudhry, R.; Ravichandran, A.; Hager, G.; Vidal, R. Histograms of oriented optical flow and Binet–Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1932–1939.
47. Bobick, A.F.; Davis, J.W. The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 257–267. [[CrossRef](#)]
48. Bobick, A.; Davis, J. An appearance-based representation of action. In Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96), Washington, DC, USA, 25–30 August, 1996; pp. 307–312.
49. Schuldts, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04), Washington, DC, USA, 23–26 August 2004; pp. 32–36.
50. Laptev, I. On space-time interest points. *Int. J. Comput. Vis.* **2005**, *64*, 107–123. [[CrossRef](#)]
51. Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*; Cambridge University Press: Cambridge, UK, 2000.
52. Vapnik, V.N. An overview of statistical learning theory. *IEEE Trans. Neural Netw.* **1999**, *10*, 988–999. [[CrossRef](#)] [[PubMed](#)]
53. Wallraven, C.; Caputo, B.; Graf, A. Recognition with local features: The kernel recipe. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 3–16 October 2003; p. 257.
54. Wolf, L.; Shashua, A. Kernel principal angles for classification machines with applications to image sequence interpretation. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 8–20 June 2003.
55. Niebles, J.C.; Fei-Fei, L. A hierarchical model of shape and appearance for human action classification. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
56. Bouchard, G.; Triggs, B. Hierarchical part-based visual object categorization. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June 2005; pp. 710–715.
57. Bosch, A.; Zisserman, A.; Munoz, X. Representing shape with a spatial pyramid kernel. In Proceedings of the 6th ACM International Conference on Image and Video Retrieval, Amsterdam, The Netherlands, 9–11 July 2007; pp. 401–408.
58. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 7–22 June 2006; pp. 2169–2178.
59. Marszałek, M.; Schmid, C.; Harzallah, H.; Van De Weijer, J. Learning object representations for visual object class recognition. In Proceedings of the Visual Recognition Challenge Workshop, in Conjunction with ICCV, Rio de Janeiro, Brazil, October 2007. Available online: <https://hal.inria.fr/inria-00548669/> (accessed on 15 July 2019).
60. Zhang, J.; Marszałek, M.; Lazebnik, S.; Schmid, C. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vis.* **2007**, *73*, 213–238. [[CrossRef](#)]
61. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the 4th Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 10–5244.
62. Horn, B.K.; Schunck, B.G. Determining optical flow. *Artif. Intell.* **1981**, *17*, 185–203. [[CrossRef](#)]
63. Chen, C.C.; Aggarwal, J. Recognizing human action from a far field of view. In Proceedings of the 2009 Workshop on Motion and Video Computing (WMVC), Snowbird, UT, USA, 8–9 December 2009; pp. 1–7.
64. Blank, M.; Gorelick, L.; Shechtman, E.; Irani, M.; Basri, R. Actions as space-time shapes. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, 17–21 October 2005; pp. 1395–1402.
65. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 886–893.

66. Hatun, K.; Duygulu, P. Pose sentences: a new representation for action recognition using sequence of pose words. In Proceedings of the 2008 19th International Conference on Pattern Recognition, Tampa, FL, USA, 8–11 December 2008; pp. 1–4.
67. Li, X. HMM based action recognition using oriented histograms of optical flow field. *Electron. Lett.* **2007**, *43*, 560–561. [[CrossRef](#)]
68. Lu, W.L.; Little, J.J. Simultaneous tracking and action recognition using the PCA-HOG descriptor. In Proceedings of the 3rd Canadian Conference on Computer and Robot Vision (CRV'06), Quebec City, QC, Canada, 7–9 June 2006; p. 6.
69. Thureau, C. Behavior histograms for action recognition and human detection. In *Human Motion—Understanding, Modeling, Capture and Animation*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 299–312.
70. Santiago-Mozos, R.; Leiva-Murillo, J.M.; Pérez-Cruz, F.; Artes-Rodriguez, A. Supervised-PCA and SVM classifiers for object detection in infrared images. In Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, Washington, DC, USA, 21–22 July 2003; pp. 122–127.
71. Chang, C.C.; Lin, C.J. LIBSVM: a library for support vector machines. *Acm Trans. Intell. Syst. Technol. TIST* **2011**, *2*, 27. [[CrossRef](#)]
72. Vishwanathan, S.; Smola, A.J.; Vidal, R. Binet–Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. *Int. J. Comput. Vis.* **2007**, *73*, 95–119. [[CrossRef](#)]
73. Schölkopf, B.; Smola, A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2002.
74. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. 1981. Available online: [https://www.researchgate.net/publication/215458777\\_An\\_Iterative\\_Image\\_Registration\\_Technique\\_with\\_an\\_Application\\_to\\_Stereo\\_Vision\\_IJCAI](https://www.researchgate.net/publication/215458777_An_Iterative_Image_Registration_Technique_with_an_Application_to_Stereo_Vision_IJCAI) (accessed on 15 July 2019).
75. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. [[CrossRef](#)]
76. Wang, H.; Schmid, C. Action Recognition with Improved Trajectories. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013.
77. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin/Heidelberg, Germany; pp. 404–417.
78. Farneback, G. Two-frame motion estimation based on polynomial expansion. In Proceedings of the Scandinavian Conference on Image Analysis, Halmstad, Sweden, 29 June–2 July 2003; Springer: Berlin/Heidelberg, Germany; pp. 363–370.
79. Prest, A.; Schmid, C.; Ferrari, V. Weakly supervised learning of interactions between humans and objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 601–614. [[CrossRef](#)] [[PubMed](#)]
80. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [[CrossRef](#)] [[PubMed](#)]
81. Akpınar, S.; Alpaslan, F.N. Video action recognition using an optical flow based representation. In Proceedings of the IPCV'14—The 2014 International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, NV, USA, 21–24 July 2014; p. 1.
82. Shi, J.; Tomasi, C. *Good Features to Track*; Technical Report; Cornell University: Ithaca, NY, USA, 1993.
83. Efros, A.A.; Berg, A.C.; Mori, G.; Malik, J. Recognizing action at a distance. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; p. 726.
84. Tran, D.; Sorokin, A. Human activity recognition with metric learning. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 548–561.
85. Ercis, F. Comparison of Histogram of Oriented Optical Flow Based Action Recognition Methods. Ph.D. Thesis, Middle East Technical University, Ankara, Turkey, 2012.
86. Li, H.; Achim, A.; Bull, D.R. GMM-based efficient foreground detection with adaptive region update. In Proceedings of the 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 3181–3184.
87. Sehgal, S. Human Activity Recognition Using BPNN Classifier on HOG Features. In Proceedings of the 2018 International Conference on Intelligent Circuits and Systems (ICICS), Phagwara, India, 19–20 April 2018; pp. 286–289.

88. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.
89. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
90. Marszałek, M.; Laptev, I.; Schmid, C. Actions in context. In Proceedings of the CVPR 2009-IEEE Conference on Computer Vision & Pattern Recognition, Miami Beach, FL, USA, 20–25 June 2009; pp. 2929–2936.
91. Niebles, J.C.; Chen, C.W.; Fei-Fei, L. Modeling temporal structure of decomposable motion segments for activity classification. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 392–405.
92. Zhang, Z. Microsoft kinect sensor and its effect. *IEEE Multimed.* **2012**, *19*, 4–10. [[CrossRef](#)]
93. Keselman, L.; Iselin Woodfill, J.; Grunnet-Jepsen, A.; Bhowmik, A. Intel realsense stereoscopic depth cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 1–10.
94. Chen, J.; Wang, B.; Zeng, H.; Cai, C.; Ma, K.K. Sum-of-gradient based fast intra coding in 3D-HEVC for depth map sequence (SOG-FDIC). *J. Vis. Commun. Image Represent.* **2017**, *48*, 329–339. [[CrossRef](#)]
95. Liang, B.; Zheng, L. A survey on human action recognition using depth sensors. In Proceedings of the 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Adelaide, SA, Australia, 23–25 November 2015; pp. 1–8.
96. Chen, C.; Liu, K.; Kehtarnavaz, N. Real-time human action recognition based on depth motion maps. *J. Real-Time Image Process.* **2016**, *12*, 155–163. [[CrossRef](#)]
97. El Madany, N.E.D.; He, Y.; Guan, L. Human action recognition via multiview discriminative analysis of canonical correlations. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 5–28 September 2016; pp. 4170–4174.
98. Yang, X.; Zhang, C.; Tian, Y. Recognizing actions using depth motion maps-based histograms of oriented gradients. In Proceedings of the 20th ACM international conference on Multimedia, Nara, Japan, 29 October–2 November 2012; ACM: New York, NY, USA, 2012; pp. 1057–1060.
99. Oreifej, O.; Liu, Z. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 716–723.
100. Wang, J.; Liu, Z.; Wu, Y.; Yuan, J. Mining actionlet ensemble for action recognition with depth cameras. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1290–1297.
101. Wang, J.; Liu, Z.; Chorowski, J.; Chen, Z.; Wu, Y. Robust 3D action recognition with random occupancy patterns. In *Computer Vision—ECCV 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 872–885.
102. Liu, M.; Liu, H.; Chen, C. Robust 3D action recognition through sampling local appearances and global distributions. *IEEE Trans. Multimed.* **2018**, *20*, 1932–1947. [[CrossRef](#)]
103. Seo, H.J.; Milanfar, P. Action recognition from one example. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 867–882. [[PubMed](#)]
104. Satyamurthi, S.; Tian, J.; Chua, M.C.H. Action recognition using multi-directional projected depth motion maps. *J. Ambient. Intell. Humaniz. Comput.* **2018**, 1–7. [[CrossRef](#)]
105. Ojala, T.; Pietikäinen, M.; Mäenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, 971–987. [[CrossRef](#)]
106. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [[CrossRef](#)]
107. Li, W.; Zhang, Z.; Liu, Z. Action recognition based on a bag of 3D points. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, San Francisco, CA, USA, 13–18 June 2010; pp. 9–14.
108. Kurakin, A.; Zhang, Z.; Liu, Z. A real time system for dynamic hand gesture recognition with a depth sensor. In Proceedings of the 20th European signal processing conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012; pp. 1975–1979.

109. Xia, L.; Chen, C.C.; Aggarwal, J.K. View invariant human action recognition using histograms of 3D joints. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 20–27.
110. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale Video Classification with Convolutional Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
111. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
112. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
113. Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1915–1929. [[CrossRef](#)]
114. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
115. Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: An astounding baseline for recognition. *arXiv* **2014**, arXiv:1403.6382.
116. Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Mao, M.; Senior, A.; Tucker, P.; Yang, K.; Le, Q.V.; et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2012; pp. 1223–1231.
117. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2014; pp. 568–576.
118. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
119. Crammer, K.; Singer, Y. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.* **2001**, *2*, 265–292.
120. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
121. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
122. Zaremba, W.; Sutskever, I. Learning to execute. *arXiv* **2014**, arXiv:1410.4615.
123. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y. Towards good practices for very deep two-stream convNets. *arXiv* **2015**, arXiv:1507.02159.
124. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
125. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
126. Wang, L.; Qiao, Y.; Tang, X. Action recognition with trajectory-pooled deep-convolutional descriptors. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 4305–4314.
127. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
128. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [[CrossRef](#)] [[PubMed](#)]
129. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941.

130. Yue-Hei Ng, J.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; Toderici, G. Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4694–4702.
131. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 20–36.
132. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
133. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
134. Sun, L.; Jia, K.; Yeung, D.Y.; Shi, B.E. Human action recognition using factorized spatio-temporal convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4597–4605.
135. Bilen, H.; Fernando, B.; Gavves, E.; Vedaldi, A.; Gould, S. Dynamic image networks for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3034–3042.
136. Fernando, B.; Gavves, E.; Oramas, J.M.; Ghodrati, A.; Tuytelaars, T. Modeling video evolution for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5378–5387.
137. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 675–678.
138. Carreira, J.; Zisserman, A. Quo vadis, action recognition? A new model and the Kinetics dataset. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4724–4733.
139. Varol, G.; Laptev, I.; Schmid, C. Long-term temporal convolutions for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1510–1517. [[CrossRef](#)]
140. Taylor, G.W.; Fergus, R.; LeCun, Y.; Bregler, C. Convolutional learning of spatio-temporal features. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 140–153.
141. Ullah, A.; Ahmad, J.; Muhammad, K.; Sajjad, M.; Baik, S.W. Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. *IEEE Access* **2018**, *6*, 1155–1166. [[CrossRef](#)]
142. Graves, A.; Fernández, S.; Schmidhuber, J. Bidirectional LSTM networks for improved phoneme classification and recognition. In Proceedings of the International Conference on Artificial Neural Networks, Warsaw, Poland, 11–15 September 2005; pp. 799–804.
143. Wang, J.; Cherian, A.; Porikli, F.; Gould, S. Video representation learning using discriminative pooling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1149–1158.
144. Schindler, K.; Van Gool, L. Action snippets: How many frames does human action recognition require? In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA, 24–26 June 2008.
145. Wang, X.; Gao, L.; Wang, P.; Sun, X.; Liu, X. Two-stream 3D convNet fusion for action recognition in videos with arbitrary size and length. *IEEE Trans. Multimed.* **2018**, *20*, 634–644. [[CrossRef](#)]
146. Liu, J.; Luo, J.; Shah, M. Recognizing realistic actions from videos in the wild. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.
147. Wang, X.; Farhadi, A.; Gupta, A. Actions~transformations. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2658–2667.
148. Chaquet, J.M.; Carmona, E.J.; Fernández-Caballero, A. A survey of video datasets for human action and activity recognition. *Comput. Vis. Image Underst.* **2013**, *117*, 633–659. [[CrossRef](#)]
149. UCF101. Action Recognition Data Set. Available online: <https://www.crcv.ucf.edu/data/UCF101.php> (accessed on 15 July 2019).



150. UCF50. Action Recognition Data Set. Available online: <https://www.crcv.ucf.edu/data/UCF50.php> (accessed on 15 July 2019).
151. HMDB: A large human motion database. Available online: <http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/> (accessed on 15 July 2019).
152. Actions as Space-Time Shapes. Available online: <http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html> (accessed on 15 July 2019).
153. MSR Action Recognition Dataset. Available online: <http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/> (accessed on 15 July 2019).
154. Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; Carlos Niebles, J. ActivityNet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 961–970.
155. A Large-Scale Video Benchmark for Human Activity Understanding. Available online: <http://activity-net.org/> (accessed on 15 July 2019).
156. Goyal, R.; Kahou, S.E.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Freund, I.; Yanilos, P.; Mueller-Freitag, M.; et al. The “Something Something” Video Database for Learning and Evaluating Visual Common Sense. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; Volume 1, p. 3.
157. The 20BN-something-something Dataset V2. Available online: <https://20bn.com/datasets/something-something> (accessed on 15 July 2019).
158. The Sports-1M Dataset. Available online: <https://github.com/gtoderic/sports-1m-dataset/blob/wiki/ProjectHome.md> (accessed on 15 July 2019).
159. YouTube-8M: A Large and Diverse Labeled Video Dataset for Video Understanding Research. Available online: <https://research.google.com/youtube8m/> (accessed on 15 July 2019).
160. Gu, C.; Sun, C.; Ross, D.A.; Vondrick, C.; Pantofaru, C.; Li, Y.; Vijayanarasimhan, S.; Toderici, G.; Ricco, S.; Sukthankar, R.; et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6047–6056.
161. AVA: A Video Dataset of Atomic Visual Action. Available online: <https://research.google.com/ava/explore.html> (accessed on 15 July 2019).
162. Lan, Z.; Lin, M.; Li, X.; Hauptmann, A.G.; Raj, B. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 204–212.
163. A Universal Labeling Tool: Sloth. Available online: <https://cvhci.anthropomatik.kit.edu/~baeu/ml/projects/a-universal-labeling-tool-for-computer-vision-sloth/> (accessed on 15 July 2019).
164. Russell, B.C.; Torralba, A.; Murphy, K.P.; Freeman, W.T. LabelMe: A Database and Web-Based Tool for Image Annotation. *Int. J. Comput. Vis.* **2008**, *77*, 157–173. [[CrossRef](#)]
165. LabelMe. Available online: <http://labelme.csail.mit.edu/Release3.0/> (accessed on 15 July 2019).
166. LabelBox. Available online: <https://labelbox.com/> (accessed on 15 July 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



# Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns

<b>Title:</b>	Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns
<b>Authors:</b>	I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, I. Rodríguez-Rodríguez, B. Sierra
<b>Journal:</b>	Sensors
<b>Publisher:</b>	MDPI
<b>DOI:</b>	10.3390/s20082436
<b>Year:</b>	2020
<b>Times cited:</b>	13 (Google Scholar) / 8 (Scopus)
<b>Source of impact:</b>	WOS (JCR)
<b>Category:</b>	ENGINEERING, ELECTRICAL & ELECTRONIC
<b>Impact index:</b>	3.576 (Q2)
<b>Position:</b>	82/273



Article

# Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns

Itsaso Rodríguez-Moreno \*, José María Martínez-Otzeta , Izaro Goienetxea, Igor Rodríguez-Rodríguez and Basilio Sierra 

Department of Computer Science and Artificial Intelligence, University of the Basque Country, Manuel Lardizabal 1, 20018 Donostia-San Sebastián, Spain; josemaria.martinezo@ehu.eus (J.M.M.-O.); izaro.goienetxea@ehu.eus (I.G.); igor.rodriguez@ehu.eus (I.R.-R.); b.sierra@ehu.eus (B.S.)

\* Correspondence: itsaso.rodriguez@ehu.eus

Received: 26 March 2020; Accepted: 23 April 2020; Published: 24 April 2020



**Abstract:** Action recognition in robotics is a research field that has gained momentum in recent years. In this work, a video activity recognition method is presented, which has the ultimate goal of endowing a robot with action recognition capabilities for a more natural social interaction. The application of Common Spatial Patterns (CSP), a signal processing approach widely used in electroencephalography (EEG), is presented in a novel manner to be used in activity recognition in videos taken by a humanoid robot. A sequence of skeleton data is considered as a multidimensional signal and filtered according to the CSP algorithm. Then, characteristics extracted from these filtered data are used as features for a classifier. A database with 46 individuals performing six different actions has been created to test the proposed method. The CSP-based method along with a Linear Discriminant Analysis (LDA) classifier has been compared to a Long Short-Term Memory (LSTM) neural network, showing that the former obtains similar or better results than the latter, while being simpler.

**Keywords:** action recognition; social robotics; common spatial patterns

---

## 1. Introduction

Social robotics aims at providing robots with artificial social intelligence to improve human–machine interaction and to introduce them in complex human contexts [1]. An effective social interaction between humans and robots requires these robots to understand and adapt to the human behaviour. Using visual perception for human activity recognition will aid the robot to provide better responses and thus enhance its social capabilities. The robot will be able to understand when the user wants to engage with it by recognising the action she/he performs.

Human activity recognition in videos is a task which consists in recognising certain actions from a series of observations. This field of research has received great attention since 1980 due to the amount of applications for which it is useful, such as health sciences, human-computer interaction, surveillance or sociology [2]. For example, in the field of surveillance [3], the automatic detection of suspicious actions allows an alert to be sent and some measures to be taken to deal with the danger. Another example is the use of action recognition for rehabilitation, which involves recognising the action the patients perform and being able to determine if they are performing it correctly or incorrectly. The principal field where this task is studied is in computer vision, based on videos. The visual features of a video provide basic information of the events or actions that occur.

Understanding what is happening in a video is really challenging, and different features can be taken into account when analysing a video sequence. For example, Video Motion Detection is a

constrained approach which consists in detecting the movement in a static background. On the other hand, Video Tracking focuses on associating objects in consecutive frames, which can be difficult if the objects are moving fast in relation to the frames per second rate. Moreover, if the object in the scene must be recognised (already a challenging task), an additional complexity is added to the problem.

In the last few years many attempts to solve these problems have been made using different techniques such as Optical Flow, Hidden Markov Models (HMM) or, more recently, deep learning [4,5]. For example, the authors of [6,7] use Histograms of Optical Flow to perform recognition. However, in [8,9] the authors use the depth information obtained by depth cameras (Microsoft Kinect or Intel Realsense), due to the fact that depth images provide additional useful information about movement. The work of [10] must also be mentioned, as it is a reference for methods that use deep learning for this task. The authors propose a two-stream architecture incorporating spatial and temporal networks, which has been used in many subsequent methods.

Considering the computational cost and the complexity that come from the need of combining temporal and spatial information, the video classification problem progresses slowly when compared with image classification.

In this paper, a new approach for video action recognition is presented. The Common Spatial Pattern algorithm is used, a method normally applied in Brain Computer Interface (BCI) for EEG systems [11]. Videos are recorded and processed with OpenPose [12] software in order to obtain a sequence of skeleton data. This skeleton data corresponds to the position of the joints of the person performing the action of the video. A sequence of skeleton data is extracted from the video, and this data can be treated as a multidimensional signal. It is then filtered according to the Common Spatial Patterns (CSP) algorithm and characteristics extracted from these filtered data are used as features for a classifier. Linear Discriminant Analysis and Random Forest (RF) classifiers have been tested to build the models from the features extracted in the previous step. Variance, maximum, minimum and interquartile range (IQR) of the filtered signals have been taken as features to feed the aforementioned classifiers. The spatial filter generated by CSP is employed as a dimensionality reduction approach and can also be interpreted in EEG data analysis as a technique that sheds light on the relationships between the filtered signals, in a similar manner to Principal Component Analysis [13] (PCA), from which it is derived. While no direct visual interpretation is possible when applied to skeleton data, this dimensionality reduction technique allows for extracting the signal components which maximally discriminate between classes.

In Figure 1 an interaction example of a person with the robot is displayed. On the left, the skeleton superposed over the actual person that is interacting with the robot is shown. The skeleton contains the (X,Y) position of 25-keypoints, which include body, head and feet information. On the right, another point of view can be seen, with the expected response of the robot. A more detailed explanation about the employed human pose estimation system and the skeleton definition is provided in Section 4.1.

To apply CSP, as a first step, the skeleton of the person appearing in each frame is extracted using OpenPose, and the (X,Y) position of each of the 25 joints that OpenPose detects are used as input data to the CSP. Therefore, in the presented method, input videos are represented as frame sequences and the temporal sequence of each skeleton joint is treated as an input signal (channel) to the CSP. In Figure 2, the following data acquisition process is shown.

In order to validate the proposed CPS-based approach, an experiment is performed where it is compared with Long Short-Term Memory [14] neural networks, yielding better results.

The rest of the paper is organised as follows. First, in Section 2 some related works are mentioned in order to introduce the topic. In Section 3 a theoretical framework is presented to explain the proposed algorithm in detail.

In Section 4 the used dataset and related skeleton capture system, as well as the experimentation carried out, are explained thoroughly, and the obtained results are shown, including a comparison between the presented approach and a Keras [15] implementation of a LSTM network. A brief

introduction to LSTMs is also presented in this section. The paper concludes with the Section 5, where the conclusions from the presented work are presented and some future work is pointed out.



(a) Image captured by the robot. (b) Expected reaction of the robot.

Figure 1. Interaction example.

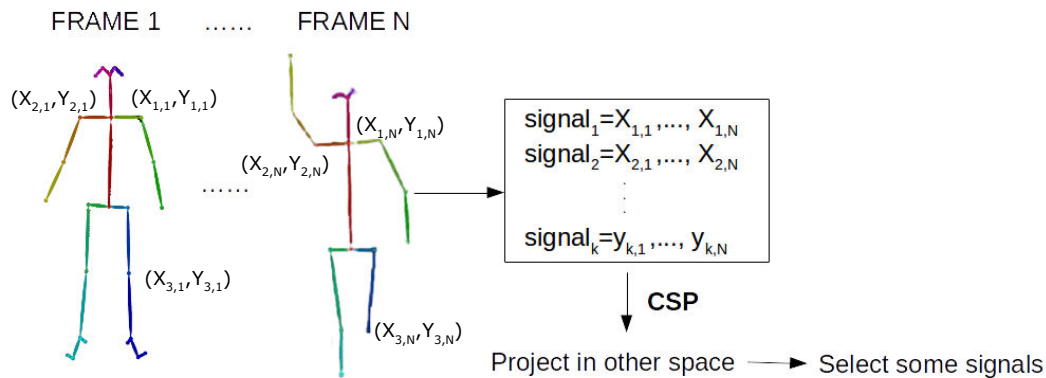


Figure 2. Proposed approach.

## 2. Related Work

As activity recognition has been an active research area lately, many different strategies have been developed to deal with this problem. There are several ways to extract visual features, both static image features and temporal visual features, and subsequently use them to perform the recognition. Temporal visual features are a combination of static image features and time information, so through these features temporal video information is achieved.

In [16] the authors use a temporal template as the basis of their representation, continuing with their work presented in [17]. This temporal template consists of a static vector-image where the value of the vector at each point represents a function of the motion properties at the corresponding spatial location in an image sequence. The authors of [18] demonstrate that local measurements in terms of spatio-temporal interest points (local features) can be used to recognise complex motion patterns. In [19] the authors present a hybrid hierarchical model, where video sequences are represented as

collections of spatial and spatio-temporal features. These features are obtained by extracting both static and dynamic interest points, and the model is able to combine static and motion image features, as well as to perform categorisation of human actions in a frame-by-frame basis. Laptev et al. [20] contribute to the recognition of realistic videos and use movie scripts for automatic annotation of human actions in videos. Due to the promising performance that they achieve in image classification [21–24], they employ spatio-temporal features and spatio-temporal pyramids, extending the spatial pyramids presented in [22].

Many other methods make use of the optical flow to solve this issue. Optical flow is the motion of objects between consecutive frames, caused by the relative movement between an observer and a scene. Therefore, optical flow methods try to calculate the motion between two image frames which are taken at times  $t$  and  $t + \Delta t$  at every position, assuming that the intensity of objects does not change during the movement. The authors of [6] use Histograms of Oriented Gradients (HOG) for human pose representations and time series of Histogram of Oriented Optical Flow (HOOF) to characterise human motion. In [7], the authors also use HOOF features for frame representation, which are independent to the scale of the moving person and to the direction of motion. There are many approaches which are based on histograms [25–27]. The authors of [28] introduce a motion descriptor based on the direction of optical flow, using the Lucas–Kanade algorithm [29] to compute it. In [30], the authors defend that to deal with the video-based action recognition problem temporally represented video information is needed. In their work, optical flow vectors are grouped according to their angular features and then summed and integrated with a new velocity concept.

It should also be mentioned that the interest of using depth data captured by depth cameras for the action recognition problem has grown, due to the advances in imaging technology to capture depth information in real time, and there are many approaches which use this extra information to make the recognition [8,9,31,32].

Some works focus on using skeleton data to perform activity recognition. In [33], the authors present a representation for action recognition, for which they use a human pose estimator and extract heatmaps for the human joints in each frame. Ren et al. [34] proposed a method for encoding geometric relational features into colour texture images, where temporal variations of different features are converted into the colour variations of their corresponding images. They use a multistream CNN model to classify the images.

As a result of the great performance that deep learning methods have achieved in image classification, these techniques have also been applied to video-based activity recognition. Taking these two publications [10,35] as a starting point, deep learning has continued to be used for activity recognition, mainly with Convolutional Neural Networks (CNN) and LSTMs. Wang et al. in their work [36] presented a very deep two-stream CNN in order to improve the results of recent architectures, getting closer to image domain deep models. In [37], trajectory-pooled deep-convolutional descriptor (TDD) is introduced, where the authors first train two-stream CNNs and then use them as feature extractors to achieve convolutional spatial and temporal feature maps from the learned networks. In the work of Feichtenhofer et al. [38], authors show that it is important to associate spatial feature maps of a particular area to temporal feature maps for that corresponding region. Authors of [39] proposed an action recognition method by processing the video data using Convolutional Neural Networks and deep bidirectional LSTM (DB-LSTM) networks. The use of deep learning for video recognition is still a work in progress, and even though the obtained results are not as good as those obtained in image recognition, better results are being achieved.

### 3. CSP-Based Approach

The core motivation of the presented method is to treat temporal sequences of skeleton joints as signals to be later processed with the CSP algorithm. In this section the CSP algorithm and the proposed approach, which makes use of that algorithm, are introduced.



### 3.1. CSP

In the last few years, the Common Spatial Pattern algorithm (first mentioned in [40] as Fukunaga-Koontz Transform) has been widely used in Brain Computer Interface (BCI) applications for electroencephalography (EEG) systems [41–43]. It is a mathematical technique used in signal processing and it consists in finding an optimum spatial filter which reduces the dimensionality of the original signals. CSP was presented as an extension of Principal Component Analysis. Considering just two different classes, a CSP filter maximises the variance of filtered signals of EEG of one of the targets while it minimises the variance for the other, in this way maximising the difference of the variances between the classes.

The feature extraction is organised in the following way:

Let  $X_1$  and  $X_2$  denote two sets of  $n$  signals where a signal is a sequence of values read from a sensor. First the covariance matrices are computed as in (1).

$$R_1 = \frac{X_1 X_1^T}{\text{trace}(X_1 X_1^T)}; \quad R_2 = \frac{X_2 X_2^T}{\text{trace}(X_2 X_2^T)} \quad (1)$$

Then, the eigen decomposition of the composite spatial covariance matrix is computed as in (2), where  $\lambda$  is the diagonal matrix of eigenvalues and  $U$  is the normalised eigenvectors matrix. To scale the principal components, the whitening transformation is used (3), obtaining an identity matrix as covariance and variance 1 for each variable.

$$R_1 + R_2 = U \lambda U^T \quad (2)$$

$$P = \sqrt{\lambda^{-1}} U^T \quad (3)$$

$R_1$  and  $R_2$  covariance matrices are transformed using  $P$  (4). After that, taking into account that the sum of two corresponding eigen values is 1 ( $\psi_1 + \psi_2 = I$ ), the eigen decomposition is computed in order to find their common eigenvectors (5).

$$S_1 = P R_1 P^T; \quad S_2 = P R_2 P^T \quad (4)$$

$$S_1 = V \psi_1 V^T; \quad S_2 = V \psi_2 V^T \quad (5)$$

The CSP filters are obtained as in (6), which maximises the separation between both classes. Using  $W$  as a projection matrix (just the first  $q$  and the last  $q$  vectors), each trial can be projected, obtaining a filtered signal matrix as in (7).

$$W = P^T V \quad (6)$$

$$Z = W^T X \quad (7)$$

The feature vector to be created for classification purposes is shown in (8), where  $\text{var}_p(Z_i)$  is the variance of the row  $p$  of the  $i$ -th trial of  $Z$ . The feature vector value for the  $p$ -th component of the  $i$ -th trial is the logarithm of the normalised variance. The feature vector has  $2q$  dimensionality, where  $q$  indicates how many vectors of the spatial filter are used in the projection. Exactly,  $q$  first and  $q$  last vectors are used, which yield the smallest variance for one class and simultaneously, the largest variance for the other class.

$$f_p^i = \log \left( \frac{\text{var}_p(Z_i)}{\sum_{p=1}^{2q} \text{var}_p(Z_i)} \right) \quad (8)$$

The purpose of this algorithm is to filter the data so their variance could be used to discriminate two populations, that is, to separate the signals belonging to two different classes. This algorithm can be useful in action recognition, where actions belonging to different classes have to be separated. From each video a group of signals is extracted (in the proposed approach, the coordinates of the joints'

positions), and then, the CSP algorithm filters the signals in a way that maximum variance difference is obtained for two different classes. Features from the filtered data obtained by CSP are therefore used as input to a classification algorithm to discriminate instances that belong to different classes.

### 3.2. Proposed Approach

Even though the CSP algorithm has been used mainly with EEG problems, in this paper a new application is presented; the use of CSP filters for feature extraction in the human action recognition task. In the presented method, each video represents a trial and each skeleton joint is treated as an EEG channel, so the videos are taken as time series where the joints of the extracted skeletons are the channels which change over time.

In Brain–Computer Interface, some electrodes are placed along the scalp and they are used to record the electrical activity of the brain. Therefore, the signals are obtained from the electrodes and then the CSP is applied using the electroencephalography signals.

However, in the proposed approach, the signals used to feed the CSP are obtained in another way. The full process can be seen in Figure 2, where the signals are composed with keypoints of the skeleton of the actor who is performing the action to recognise. Each trial is a video where the signals are the values of the skeleton position over time. Once the skeletons are processed and, hence, the signals are formed, the CSP is computed in order to separate the classes according to their variance.

The main focus of the experimentation is the use of the variance of the signals after applying the Common Spatial Pattern algorithm as input to the classification algorithms. However, in addition to the variance, much more information can be extracted from these transformed signals, which may be useful when performing the classification. Hence, some experiments are performed with just the information of the variances and other experiments also with information about the maximum, minimum and the interquartile range ( $IQR = Q3 - Q1$ ) of the signal. Once the features are extracted from the transformed signals, Linear Discriminant Analysis and Random Forest classifiers are used to perform the classification. The Linear Discriminant Analysis [44] tries to separate the different classes by finding a linear combination of features which describe each of the targets. Random Forest [45] is a Bagging (Bootstrap Aggregating) multiclassifier composed of decision trees.

## 4. Experimental Results

### 4.1. Robotic Platform and Human Pose Estimation

The robotic platform employed in the performed experiments is a Pepper robot developed by Softbank Robotics (<https://www.softbankrobotics.com/emea/en/pepper>). Pepper is a human-like torso that is fitted onto a holonomous wheeled platform. It is equipped with full-colour RGB LEDs, three cameras and several sensors located in different parts of its body that allow for perceiving the surrounding environment with high precision. In this work, only the information provided by the two identical RGB cameras, with a resolution of  $320 \times 240$  pixels, situated on the forehead of the robot has been used (see Figure 3). The images of both cameras have been combined to obtain a wider field of view and better capture the person in front of the robot, thus obtaining an image of  $320 \times 480$  resolution. An example of the combined image is shown in Figure 1a.

In order to obtain the data to apply CSP, as a first step, the skeleton of the person appearing in the scene has to be obtained. For this purpose, it has been decided to extract the skeletons using OpenPose [12], one of the most popular bottom-up approaches for multiperson human pose estimation. As with many bottom-up techniques, OpenPose first detects parts (keypoints) belonging to every person in the image and then assigns those parts to distinct individuals. The assignment is made using a nonparametric representation of association scores via Part Affinity Fields (PAFs), a set of 2d vectors fields that encode the location and orientation of limbs over the image. OpenPose can detect human body, feet, hands, and facial keypoints (135 keypoints in total) on single images. Due to the high computational cost that estimating all the keypoints requires, in this work only the BODY\_25

(COCO [46] + feet) model has been used for human pose estimation. It returns the (X,Y) positions in the image of the extracted 25-keypoints, including head, body, and feet (see Figure 4).

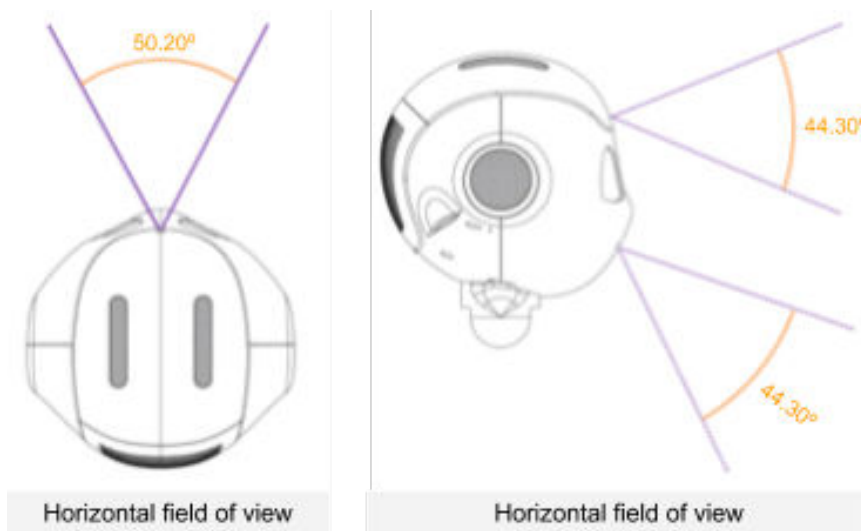


Figure 3. Pepper’s RGB cameras position and orientation.

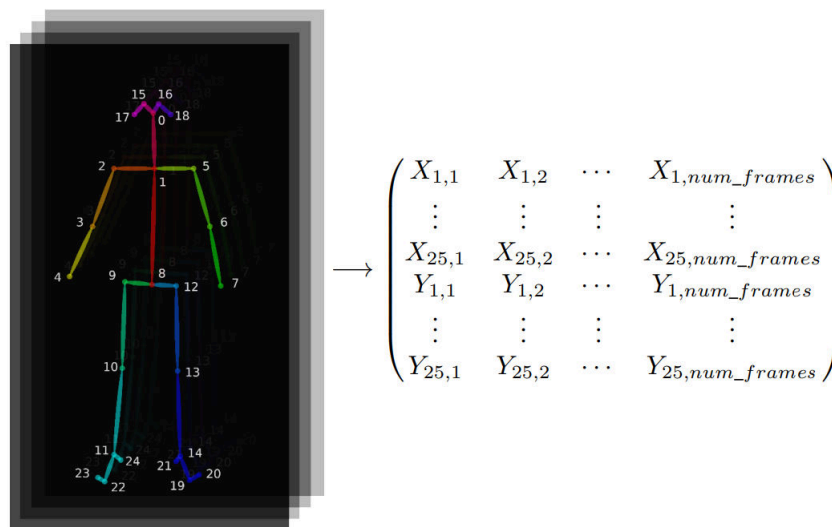


Figure 4. Skeleton’s joint positions and matrix representation of the extracted signals.

#### 4.2. Dataset

The videos in the database have been recorded using the combined image obtained from Pepper’s forehead cameras. It consists of 272 videos with six action categories and around 45 clips belong to each category, performed by 46 different people. The robot adjusts the orientation of its head according to the location of the face of the person appearing in its field of view.

All the participants in this study gave their consent in being recorded for this research purpose. No raw video data has been stored, and only minimum information about joints’ spatial coordinates has been maintained. All this data is anonymised, with no information about sex, age, race, or any other condition of the participants.

The action categories and video information can be seen in Table 1.

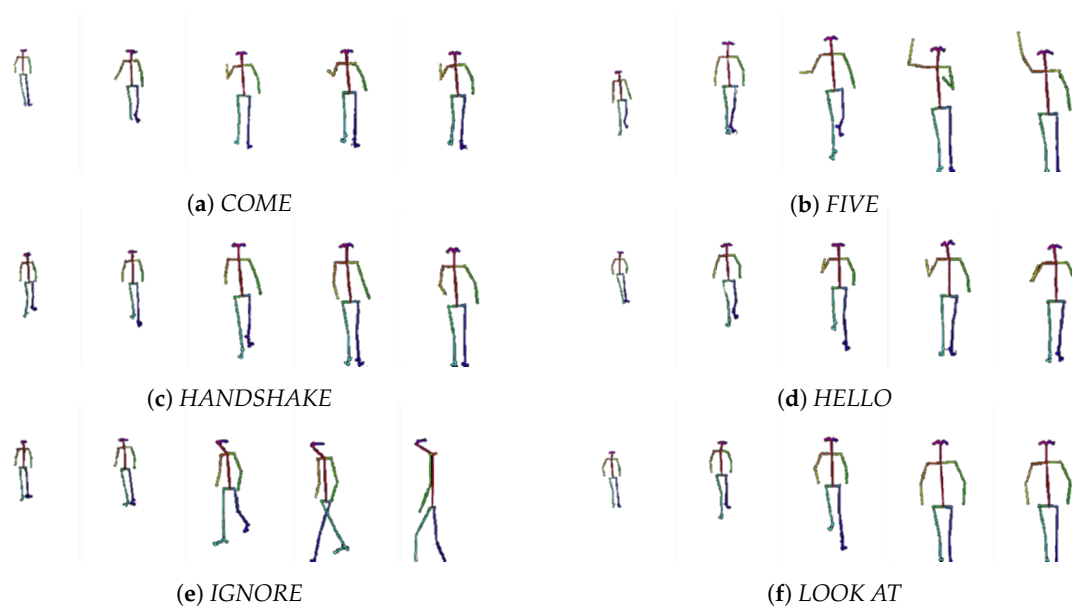
**Table 1.** Characteristics of each action category.

Category	#Video	Resolution	FPS
COME	46	320 × 480	10
FIVE	45	320 × 480	10
HANDSHAKE	45	320 × 480	10
HELLO	44	320 × 480	10
IGNORE	46	320 × 480	10
LOOK AT	46	320 × 480	10

These are the six categories that the robot must differentiate:

1. COME: gesture for telling the robot to come to you.
2. FIVE: gesture of "high five".
3. HANDSHAKE: gesture of handshaking with the robot.
4. HELLO: gesture for indicating hello to the robot.
5. IGNORE: ignore the robot, pass by.
6. LOOK AT: stare at the robot in front of it.

Examples of skeletons extracted from videos of the six different classes are shown in Figure 5. It can be seen in the examples that all the videos follow the same pattern: the actor appears in the scene, approaches the robot and finally, the action is performed.

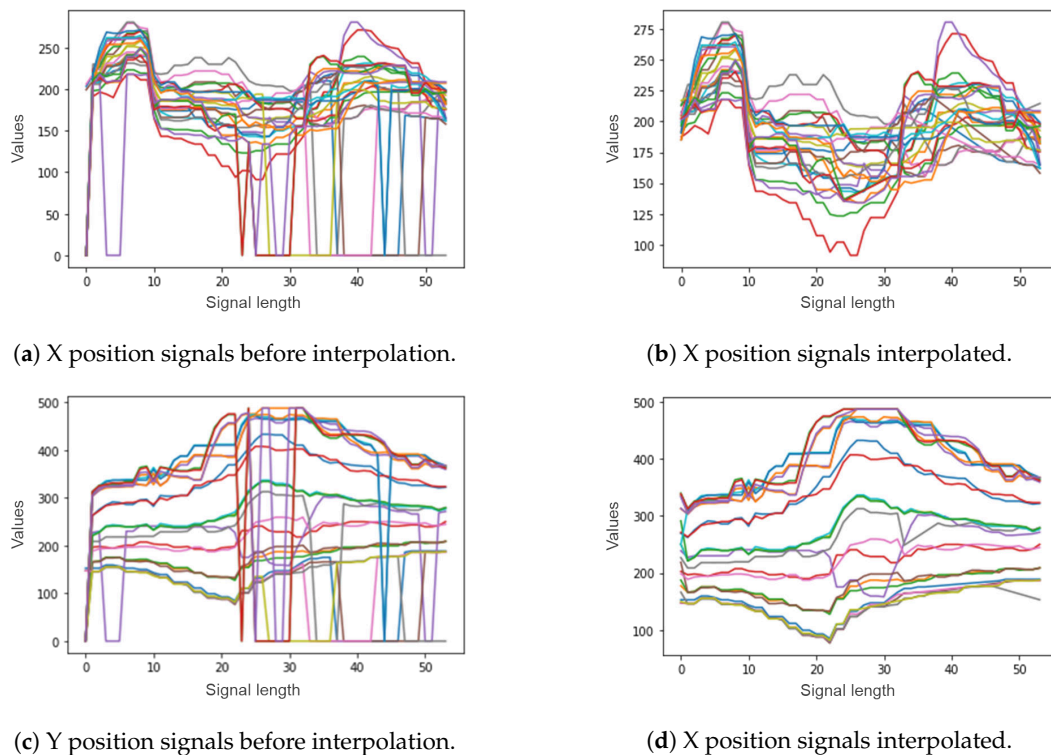


**Figure 5.** Frame sequence examples for different categories.

In this case, the actions that have to be recognised are centred in the actor who performs them. Therefore, the skeleton of the actor has been extracted in every frame of each video. OpenPose returns the (X,Y) positions of 25-keypoints (joints). After obtaining the skeleton information for every frame of each video, fifty different signals are created to represent each video, where each signal will be the position of a skeleton keypoint over time. This way, there will be 50 signals (25 for the X position of the joints and another 25 for the Y position) with the same length as the original video (one skeleton per frame). The skeleton appearance and the matrix extracted from skeletons can be seen in Figure 4.

Some joints could be missing from the captured skeletons when the actor does not fit entirely in the camera range. In these cases, the missing joint values are estimated by a linear interpolation, using the previous and next values for that joint. The interpolation is done to avoid missing values and assuming that consecutive values of joints positions follow a smooth curve. The process of interpolation for the signal of one video can be seen graphically in Figure 6, where Figure 6a,c

show the 25 X and 25 Y signals before interpolation and Figure 6b,d the 25 X and 25 Y signals after interpolating them.



**Figure 6.** Linear interpolation example.

Furthermore, the length of all the input data must be the same to apply the proposed method, therefore, it might be necessary to apply a preprocessing step to the videos. As the duration of the original videos differ, it has been decided to convert all the videos to the length of the longest clip.

As mentioned before, OpenPose provides the skeletons of the people of the scene for each frame of the video. It could happen that in some frames no person is detected and no skeleton is formed. Analysing this dataset, it can be noticed that full skeletons are only missed at the beginning of some of the videos and it has been decided to repeat the first skeleton encountered as many times as necessary.

After performing these changes, 50 signals with maximum video's length are obtained. These signals are then used to feed the CSP.

#### 4.3. Long Short-Term Memory (LSTM) Neural Networks

LSTMs are a category of recurrent neural networks (RNNs) which belong to the growing field of deep learning paradigms. RNNs are artificial neural networks in which connections between units form a directed cycle. Due to this architecture, recurrent neural networks possess an internal state that stores information about past inputs. This endows the recurrent networks with the ability to process sequences of inputs and exhibits a dynamic temporal behaviour in response to those sequences.

Training RNNs to learn long-term dependencies by gradient-descent methods used to be difficult due to the vanishing or exploding gradient problem [47,48]. In recent years, sophisticated optimisation techniques, specialised network designs, and new weight initialisation methods have addressed this problem with great success [49]. LSTM design introduces gates that control how much of the past and the current state has to get through to the next time step.

In a RNN, the following terms are defined:

- $x_t$ : input vector at time step  $t$ .
- $h_t = \phi(Wx_t + Uh_{t-1})$ : hidden state at time step  $t$ .  $W$  and  $U$  are weight matrices applied to the current input and to the previous hidden state, respectively.  $\phi$  is an activation function, typically sigmoid ( $\sigma$ ), tanh, or ReLU.
- $o_t = \text{softmax}(Vs_t)$ : output vector at time step  $t$ .  $V$  is a weight matrix.

In LSTMs accounting for the capability of forgetting selectively, the node's state is needed, so the terms are typically the following:

- $x_t$ : input vector at time step  $t$ .
- $f_t = \sigma(W_f x_t + U_f h_{t-1})$ : activation vector of the forget gate at time step  $t$ .
- $i_t = \sigma(W_i x_t + U_i h_{t-1})$ : activation vector of the input gate at time step  $t$ .
- $o_t = \sigma(W_o x_t + U_o h_{t-1})$ : activation vector of the output gate at time step  $t$ .
- $c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_c x_t + U_c h_{t-1})$ : cell state vector at time step  $t$ .
- $h_t = o_t \circ \tanh(c_t)$ : hidden state at time step  $t$ .

$W_f, W_i, W_o,$  and  $W_c$  are weight matrices applied to the current input, while  $U_f, U_i, U_o,$  and  $U_c$  are applied to the previous hidden state. The  $\circ$  operator represents the Hadamard product.

#### 4.4. Results

Once the data have been processed, the previously explained CSP algorithm is performed. The used CSP method is implemented to work with just two classes, therefore all the tests have been carried out using pairs of classes, although multiclass classification is possible using pairwise classification approaches, such as One versus One (OVO) as a class binarization technique [50].

In Table 2 the obtained results by Linear Discriminant Analysis (LDA) classifier can be seen, and in Table 3 the results obtained by RF classifier are shown, where best results are highlighted in boldface. Both tables present the accuracy values obtained for every pair of classes of the database, using 10-fold cross validation for the evaluation. Parameter  $q$  indicates that only  $2 \times q$  feature vectors are considered, where  $2 \times q$  are the  $q$  first and  $q$  last vectors, when sorted by variance. Therefore, a feature vector of  $2 \times q$  dimensionality is obtained after applying CSP, and that feature vector is the input to LDA or RF classifiers. In each table the accuracy values obtained with two different types of feature vectors are shown; variance when only the variances of the transformed signals are used to form the feature vectors and variance, max, min, IQR when apart from the variances, maximum, minimum, and IQR values are also represented in the feature vectors.

**Table 2.** Results obtained applying Common Spatial Patterns (CSP) with different  $q$  values and using LDA as classifier.

Pair of Categories	Variance			Variance, Max, Min, IQR		
	$q = 5$	$q = 10$	$q = 15$	$q = 5$	$q = 10$	$q = 15$
COME-FIVE	0.7579 ± 0.13	0.8124 ± 0.12	0.7667 ± 0.17	0.7578 ± 0.12	<b>0.8344 ± 0.14</b>	0.7667 ± 0.16
COME-HANDSHAKE	<b>0.8668 ± 0.10</b>	0.8019 ± 0.12	0.6910 ± 0.17	<b>0.8667 ± 0.13</b>	0.7900 ± 0.12	0.6567 ± 0.16
COME-HELLO	<b>0.5334 ± 0.16</b>	0.5000 ± 0.09	0.5000 ± 0.14	0.4778 ± 0.16	<b>0.4444 ± 0.09</b>	0.4778 ± 0.15
COME-IGNORE	<b>0.9779 ± 0.05</b>	0.9667 ± 0.05	0.9667 ± 0.05	0.9667 ± 0.05	0.9667 ± 0.05	0.9444 ± 0.06
COME-LOOK_AT	0.8678 ± 0.09	<b>0.8900 ± 0.09</b>	0.8789 ± 0.11	0.8678 ± 0.10	0.8356 ± 0.14	0.8033 ± 0.14
FIVE-HAND	<b>0.9557 ± 0.06</b>	0.9333 ± 0.06	0.9223 ± 0.05	0.9333 ± 0.11	0.9000 ± 0.11	0.9000 ± 0.08
FIVE-HELLO	<b>0.8208 ± 0.14</b>	0.7986 ± 0.15	0.7764 ± 0.17	0.7750 ± 0.18	0.7528 ± 0.18	0.7319 ± 0.21
FIVE-IGNORE	<b>0.9668 ± 0.07</b>	<b>0.9668 ± 0.07</b>	0.9556 ± 0.11	<b>0.9667 ± 0.07</b>	0.9556 ± 0.11	0.9556 ± 0.11
FIVE-LOOK_AT	<b>0.9667 ± 0.05</b>	0.9556 ± 0.06	0.9556 ± 0.06	0.9556 ± 0.08	0.9556 ± 0.08	0.9011 ± 0.17
HANDSHAKE-HELLO	0.7431 ± 0.19	0.7861 ± 0.14	0.8097 ± 0.10	0.7111 ± 0.24	0.7889 ± 0.21	0.8000 ± 0.10
HANDSHAKE-IGNORE	0.9889 ± 0.04	<b>1.0000 ± 0.00</b>	1.0000 ± 0.00	<b>1.0000 ± 0.00</b>	0.9889 ± 0.04	0.9889 ± 0.04
HANDSHAKE-LOOK_AT	<b>0.8235 ± 0.18</b>	0.7789 ± 0.16	0.7567 ± 0.12	0.8122 ± 0.17	0.7467 ± 0.17	0.7456 ± 0.12
HELLO-IGNORE	0.9333 ± 0.14	0.9221 ± 0.14	0.9333 ± 0.11	<b>0.9556 ± 0.14</b>	0.9444 ± 0.14	0.9444 ± 0.11
HELLO-LOOK_AT	0.8445 ± 0.11	0.8334 ± 0.12	0.8556 ± 0.14	0.8556 ± 0.09	0.8000 ± 0.10	<b>0.8667 ± 0.10</b>
IGNORE-LOOK_AT	<b>0.9889 ± 0.04</b>	<b>0.9889 ± 0.04</b>	0.9889 ± 0.04	0.9778 ± 0.05	0.9678 ± 0.05	0.9678 ± 0.05
MEAN	<b>0.8691</b>	0.8623	0.8506	0.8586	0.8448	0.8301

**Table 3.** Results obtained applying CSP with different  $q$  values and using RF as classifier.

Pair of Categories	Variance			Variance, Max, Min, IQR		
	$q = 5$	$q = 10$	$q = 15$	$q = 5$	$q = 10$	$q = 15$
COME-FIVE	0.6800 ± 0.29	0.6022 ± 0.24	0.5811 ± 0.19	<b>0.7133 ± 0.21</b>	0.6244 ± 0.23	0.5922 ± 0.21
COME-HANDSHAKE	0.7000 ± 0.20	0.6900 ± 0.29	0.6344 ± 0.29	<b>0.7556 ± 0.16</b>	0.6678 ± 0.32	0.6344 ± 0.32
COME-HELLO	<b>0.5111 ± 0.22</b>	0.3889 ± 0.21	0.4222 ± 0.17	0.4889 ± 0.22	0.4222 ± 0.20	0.3889 ± 0.20
COME-IGNORE	<b>0.9233 ± 0.12</b>	0.8900 ± 0.17	0.8800 ± 0.18	<b>0.9233 ± 0.12</b>	0.8911 ± 0.15	0.8578 ± 0.20
COME-LOOK_AT	<b>0.8133 ± 0.23</b>	0.7800 ± 0.20	0.7456 ± 0.25	0.8122 ± 0.23	0.8122 ± 0.24	0.7789 ± 0.24
FIVE-HANDSHAKE	<b>0.8889 ± 0.17</b>	0.7778 ± 0.15	0.6444 ± 0.17	0.8444 ± 0.17	0.7667 ± 0.12	0.6667 ± 0.17
FIVE-HELLO	<b>0.6264 ± 0.22</b>	0.5500 ± 0.22	0.5028 ± 0.23	<b>0.6264 ± 0.22</b>	0.5361 ± 0.23	0.5236 ± 0.24
FIVE-IGNORE	0.9444 ± 0.14	0.9344 ± 0.14	0.9344 ± 0.14	<b>0.9556 ± 0.11</b>	0.9456 ± 0.11	0.9233 ± 0.14
FIVE-LOOK_AT	0.9000 ± 0.19	0.8889 ± 0.21	0.8233 ± 0.23	<b>0.9111 ± 0.21</b>	0.9000 ± 0.21	0.8556 ± 0.25
HANDSHAKE-HELLO	0.6875 ± 0.18	0.5708 ± 0.14	0.6111 ± 0.20	<b>0.6889 ± 0.19</b>	0.5819 ± 0.16	0.6556 ± 0.15
HANDSHAKE-IGNORE	<b>0.9789 ± 0.04</b>	0.9578 ± 0.07	0.9133 ± 0.12	<b>0.9789 ± 0.04</b>	0.9578 ± 0.07	0.9244 ± 0.11
HANDSHAKE-LOOK_AT	0.7344 ± 0.26	<b>0.7556 ± 0.29</b>	0.6789 ± 0.29	0.7456 ± 0.26	0.7456 ± 0.28	0.6678 ± 0.25
HELLO-IGNORE	0.9000 ± 0.14	0.8889 ± 0.17	0.8667 ± 0.21	<b>0.9111 ± 0.15</b>	0.8889 ± 0.17	0.8667 ± 0.21
HELLO-LOOK_AT	0.7667 ± 0.22	0.6556 ± 0.32	0.6556 ± 0.35	<b>0.7889 ± 0.23</b>	0.7556 ± 0.29	0.7333 ± 0.28
IGNORE-LOOK_AT	0.9222 ± 0.12	<b>0.9333 ± 0.14</b>	0.9222 ± 0.14	<b>0.9333 ± 0.09</b>	0.9111 ± 0.15	<b>0.9333 ± 0.14</b>
MEAN	0.7985	0.7509	0.7211	<b>0.8052</b>	0.7605	0.7335

Looking at the results of Table 2, it can be observed that best outcomes are achieved when  $q = 5$ , that is, taking 10 values per video is enough to perform the classification. An accuracy higher than 80% is attained for most of the category pairs. Regarding the categories, some of them are better distinguished than others. For example, good results are obtained when classifying the class ignore with all other classes, so it can be supposed that the features obtained for the category ignore are quite different from the rest. However, videos that belong to the pair of classes come and hello are more difficult to differentiate, which can be easily deduced looking at the skeletons of both classes. Concerning the feature vector type, the results indicate that there is no need to use more information than the variances of the transformed signals to obtain better results; the accuracy values obtained with the variances are higher. Nevertheless, the obtained results indicate that the presented approach yields a good classification accuracy.

The results of Table 3 show that RF classifier performs worse than LDA, obtaining lower accuracy values in general. In this case, the feature vector type which uses the variance, max, min, and IQR values achieves better outcomes. Regarding both the  $q$  value and the categories, the conclusions presented for the results obtained by LDA classifier are maintained.

In order to assess the effectiveness of the presented method when compared with another technique, a Long Short-Term Memory network has been chosen, as this type of neural network has been widely used for video action recognition tasks. The LSTM network has been implemented in Python using the Keras library. The input shape is bidimensional (number of frames, number of joints), and the output space is of 64 units. Then another dense layer for classification is added, of size 2, as this is the number of classes for each individual problem. The Adam optimisation algorithm [51] has been used, as well as categorical cross-entropy as loss function. It has been trained during 100 epochs, with a batch size of 25. The comparison is made between the aforementioned LSTM and the proposed approach with the configuration which has achieved highest accuracy, in this instance, variance  $q = 5$  with LDA classifier. The results are shown in Table 4, where best results are highlighted in boldface.

LSTM achieves accuracy values between 70% and 90% for most of the pairs. In this case, the accuracy obtained for come-hello pair has been improved notoriously. However, the results obtained for the rest of the classes are not that significant.

The results show that the presented method performs better than LSTM. More precisely, it outperforms LSTM results for 9 of 15 category pairs. Moreover, the mean value of all the tested pairs has been calculated for each technique, and it can be concluded that the proposed approach obtains higher accuracy values. Therefore, the CSP-based method not only achieves better results in most classifications but the average of the values obtained is higher.

**Table 4.** Comparison between the proposed approach and LSTM approach.

Pair of Categories	CSP (Variance and $q = 5$ ) + LDA	LSTM
COME-FIVE	0.7579 ± 0.13	<b>0.8628 ± 0.11</b>
COME-HANDSHAKE	<b>0.8668 ± 0.10</b>	0.7739 ± 0.16
COME-HELLO	0.5334 ± 0.16	<b>0.7336 ± 0.17</b>
COME-IGNORE	<b>0.9779 ± 0.05</b>	0.9575 ± 0.06
COME-LOOK_AT	<b>0.8678 ± 0.09</b>	0.7849 ± 0.10
FIVE-HANDSHAKE	<b>0.9557 ± 0.06</b>	0.8125 ± 0.14
FIVE-HELLO	0.8208 ± 0.14	<b>0.9125 ± 0.07</b>
FIVE-IGNORE	0.9668 ± 0.07	<b>0.9789 ± 0.04</b>
FIVE-LOOK_AT	<b>0.9667 ± 0.05</b>	0.8889 ± 0.11
HANDSHAKE-HELLO	<b>0.7431 ± 0.19</b>	0.7108 ± 0.21
HANDSHAKE-IGNORE	<b>0.9889 ± 0.04</b>	0.9764 ± 0.05
HANDSHAKE-LOOK_AT	0.8235 ± 0.18	<b>0.8350 ± 0.12</b>
HELLO-IGNORE	0.9333 ± 0.14	<b>0.9789 ± 0.04</b>
HELLO-LOOK_AT	<b>0.8445 ± 0.11</b>	0.5733 ± 0.18
IGNORE-LOOK_AT	<b>0.9889 ± 0.04</b>	0.9775 ± 0.05
MEAN	<b>0.8691</b>	0.8505

Furthermore, the other three configurations tested above with LDA classifier (variance- $q = 10$ , variance- $q = 15$  and variance, max, min, IQR- $q = 5$ ) also outperform the results obtained by the LSTM method.

$$\begin{array}{ccccccc} \text{variance } q = 5 & & \text{variance } q = 10 & & \text{var, max, min, IQR } q = 5 & & \text{variance } q = 15 & & \text{LSTM} \\ 0.8691 & > & 0.8622 & > & 0.8586 & > & 0.8506 & > & 0.8505 \end{array}$$

## 5. Conclusions

In this paper a new approach for activity recognition in video sequences is presented, in which Common Spatial Pattern signal processing has been applied to the skeleton joints data of people performing different activities. Features extracted from the transformed data have been used as input to Linear Discriminant Analysis and Random Forest classifiers, in order to perform action recognition. Two different sets of features have been selected: {Variance} and {Variance, Max, Min, IQR}. The results show that CSP processing followed by LDA classifier over variance features compares favourably to a Long Short-Term Memory model trained with the same data. From a database of six actions (fifteen possible pairs of actions), CSP and LDA obtains better results than LSTM in 9 of 15 category pairs.

Another advantage of the proposed method is the relative simplicity of LDA compared to LSTM networks and the lack of need for hyperparameter tuning. The set of features is also small, since only variance is used in the model that achieves best results.

As further work, it is planned to extend the range of human activities. Implementation of a real-time system could be of interest, for example, in social robotics.

**Author Contributions:** Research concept and supervision of technical writing: B.S.; software implementation, concept development, and technical writing: I.R.-M.; results validation and supervision of technical writing: J.M.M.-O. and I.G.; methodological analysis: I.R.-R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially funded by the Basque Government, Spain, grant number IT900-16, and the Spanish Ministry of Science (MCIU), the State Research Agency (AEI), and the European Regional Development Fund (FEDER), grant number RTI2018-093337-B-I00 (MCIU/AEI/FEDER, UE).

**Acknowledgments:** We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Breazeal, C. *Designing Sociable Robots*; Intelligent Robotics and Autonomous Agents, MIT Press: Cambridge, MA, USA, 2004.
2. Ke, S.R.; Thuc, H.; Lee, Y.J.; Hwang, J.N.; Yoo, J.H.; Choi, K.H. A review on video-based human activity recognition. *Computers* **2013**, *2*, 88–131. [[CrossRef](#)]



3. Vishwakarma, S.; Agrawal, A. A survey on activity recognition and behavior understanding in video surveillance. *Vis. Comput.* **2013**, *29*, 983–1009. [[CrossRef](#)]
4. Poppe, R. A survey on vision-based human action recognition. *Image Vis. Comput.* **2010**, *28*, 976–990. [[CrossRef](#)]
5. Herath, S.; Harandi, M.; Porikli, F. Going deeper into action recognition: A survey. *Image Vis. Comput.* **2017**, *60*, 4–21. [[CrossRef](#)]
6. Chen, C.C.; Aggarwal, J. Recognizing human action from a far field of view. In Proceedings of the 2009 Workshop on Motion and Video Computing (WMVC), Snowbird, UT, USA, 8–9 December 2009; pp. 1–7.
7. Chaudhry, R.; Ravichandran, A.; Hager, G.; Vidal, R. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1932–1939.
8. Oreifej, O.; Liu, Z. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 716–723.
9. Liu, M.; Liu, H.; Chen, C. Robust 3D action recognition through sampling local appearances and global distributions. *IEEE Trans. Multimed.* **2018**, *20*, 1932–1947. [[CrossRef](#)]
10. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*; The MIT Press, Cambridge, MA, USA, 2014; pp. 568–576.
11. Astigarraga, A.; Arruti, A.; Muguerza, J.; Santana, R.; Martin, J.I.; Sierra, B. User adapted motor-imaginary brain-computer interface by means of EEG channel selection based on estimation of distributed algorithms. *Math. Probl. Eng.* **2016**, 2016. [[CrossRef](#)]
12. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv* **2018**, arXiv:1812.08008.
13. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **1933**, *24*, 417. [[CrossRef](#)]
14. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
15. Chollet, F. Keras. 2015. Available online: <https://keras.io> (accessed on 24 April 2020).
16. Bobick, A.F.; Davis, J.W. The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 257–267. [[CrossRef](#)]
17. Bobick, A.; Davis, J. An appearance-based representation of action. In Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, 25–19 August 1996; Volume 1; pp. 307–312.
18. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 26 August 2004, Volume 3; pp. 32–36.
19. Niebles, J.C.; Fei-Fei, L. A hierarchical model of shape and appearance for human action classification. In Proceedings of the Computer Vision and Pattern Recognition, CVPR'07, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
20. Laptev, I.; Marszalek, M.; Schmid, C.; Rozenfeld, B. Learning realistic human actions from movies. In Proceedings of the Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
21. Bosch, A.; Zisserman, A.; Munoz, X. Representing shape with a spatial pyramid kernel. In Proceedings of the 6th ACM International Conference on Image And Video Retrieval, Amsterdam, The Netherlands, 9–11 July 2007; pp. 401–408.
22. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; pp. 2169–2178.
23. Marszałek, M.; Schmid, C.; Harzallah, H.; Van De Weijer, J. Learning object representations for visual object class recognition. In Proceedings of the Visual Recognition Challenge Workshop, in Conjunction with ICCV, Rio de Janeiro, Brazil, 14–20 October 2007.
24. Zhang, J.; Marszałek, M.; Lazebnik, S.; Schmid, C. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Comput. Vis.* **2007**, *73*, 213–238. [[CrossRef](#)]

25. Efros, A.A.; Berg, A.C.; Mori, G.; Malik, J. Recognizing action at a distance. In Proceedings of the Ninth International Conference on Computer Vision, Nice, France, 13–16 October 2003; pp. 726–733.
26. Tran, D.; Sorokin, A. Human activity recognition with metric learning. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 548–561.
27. Ercis, F. Comparison of Histogram of Oriented Optical Flow Based Action Recognition Methods. Ph.D. Thesis, Middle East Technical University, Ankara, Turkey, 2012.
28. Lertniphonphan, K.; Aramvith, S.; Chalidabhongse, T.H. Human action recognition using direction histograms of optical flow. In Proceedings of the Communications and Information Technologies (ISCIT), 2011 11th International Symposium on Communications & Information Technologies (ISCIT 2011), Hangzhou, China, 12–14 October 2011; pp. 574–579.
29. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique With an Application To Stereo Vision; In Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981; pp. 674–679.
30. Akpinar, S.; Alpaslan, F.N. Video action recognition using an optical flow based representation. In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), Las Vegas, NV, USA, 21–24 September 2014; p. 1.
31. Satyamurthi, S.; Tian, J.; Chua, M.C.H. Action recognition using multi-directional projected depth motion maps. *J. Ambient. Intell. Humaniz. Comput.* **2018**, *9*, 1–7. [[CrossRef](#)]
32. Yang, X.; Zhang, C.; Tian, Y. Recognizing actions using depth motion maps-based histograms of oriented gradients. In Proceedings of the 20th ACM International Conference on Multimedia, Nara, Japan, 29 October–2 November 2012; pp. 1057–1060.
33. Choutas, V.; Weinzaepfel, P.; Revaud, J.; Schmid, C. PoTion: Pose MoTion Representation for Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
34. Ren, J.; Reyes, N.H.; Barczak, A.; Scogings, C.; Liu, M. An Investigation of Skeleton-Based Optical Flow-Guided Features for 3D Action Recognition Using a Multi-Stream CNN Model. In Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018; pp. 199–203.
35. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
36. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y. Towards good practices for very deep two-stream ConvNets. *arXiv* **2015**, arXiv:1507.02159.
37. Wang, L.; Qiao, Y.; Tang, X. Action recognition with trajectory-pooled deep-convolutional descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4305–4314.
38. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1933–1941.
39. Ullah, A.; Ahmad, J.; Muhammad, K.; Sajjad, M.; Baik, S.W. Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. *IEEE Access* **2018**, *6*, 1155–1166. [[CrossRef](#)]
40. Fukunaga, K.; Koontz, W.L. Application of the Karhunen-Loève Expansion to Feature Selection and Ordering. *IEEE Trans. Comput.* **1970**, *100*, 311–318. [[CrossRef](#)]
41. Ramoser, H.; Muller-Gerking, J.; Pfurtscheller, G. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 441–446. [[CrossRef](#)] [[PubMed](#)]
42. Wang, Y.; Gao, S.; Gao, X. Common spatial pattern method for channel selection in motor imagery based brain-computer interface. In Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 17–18 January 2006; pp. 5392–5395.
43. Novi, Q.; Guan, C.; Dat, T.H.; Xue, P. Sub-band common spatial pattern (SBCSP) for brain-computer interface. In Proceedings of the 2007 3rd International IEEE/EMBS Conference on Neural Engineering, Kohala Coast, HI, USA, 2–5 May 2007; pp. 204–207.

44. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [[CrossRef](#)]
45. Ho, T.K. Random decision forests. In Proceedings of the 3rd international conference on document analysis and recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1; pp. 278–282.
46. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
47. Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J. Gradient flow in recurrent nets: The difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*; Kremer, S.C., Kolen, J.F., Eds.; IEEE Press: Piscataway, NJ, USA, 2001.
48. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [[CrossRef](#)]
49. Talathi, S.S.; Vartak, A. Improving performance of recurrent neural network with relu nonlinearity. *arXiv* **2015**, arXiv:1511.03771.
50. Mendialdua, I.; Martínez-Otzeta, J.M.; Rodríguez-Rodríguez, I.; Ruiz-Vazquez, T.; Sierra, B. Dynamic selection of the best base classifier in one versus one. *Knowl.-Based Syst.* **2015**, *85*, 298–306. [[CrossRef](#)]
51. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



# Using Common Spatial Patterns to Select Relevant Pixels for Video Activity Recognition

**Title:** Using Common Spatial Patterns to Select Relevant Pixels for Video Activity Recognition

**Authors:** I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra, I. Irigoien, I. Rodríguez-Rodríguez, I. Goienetxea

**Journal:** Applied Sciences

**Publisher:** MDPI

**DOI:** 10.3390/app10228075

**Year:** 2020

**Times cited:** 1 (Google Scholar)

**Source of impact:** WOS (JCR)

**Category:** ENGINEERING, MULTIDISCIPLINARY

**Impact index:** 2.679 (Q2)

**Position:** 38/90



Article

# Using Common Spatial Patterns to Select Relevant Pixels for Video Activity Recognition

Itsaso Rodríguez-Moreno \*, José María Martínez-Otzeta, Basilio Sierra, Itziar Irigoien, Igor Rodriguez-Rodriguez and Izaro Goienetxea

Department of Computer Science and Artificial Intelligence, University of the Basque Country, Manuel Lardizabal 1, 20018 Donostia-San Sebastián, Spain; josemaria.martinezo@ehu.eus (J.M.M.-O.); b.sierra@ehu.eus (B.S.); itziar.irigoien@ehu.eus (I.I.); igor.rodriguez@ehu.eus (I.R.-R.); izaro.goienetxea@ehu.eus (I.G.)

\* Correspondence: itsaso.rodriguez@ehu.eus

Received: 1 October 2020; Accepted: 11 November 2020; Published: 14 November 2020



**Abstract:** Video activity recognition, despite being an emerging task, has been the subject of important research due to the importance of its everyday applications. Video camera surveillance could benefit greatly from advances in this field. In the area of robotics, the tasks of autonomous navigation or social interaction could also take advantage of the knowledge extracted from live video recording. In this paper, a new approach for video action recognition is presented. The new technique consists of introducing a method, which is usually used in Brain Computer Interface (BCI) for electroencephalography (EEG) systems, and adapting it to this problem. After describing the technique, achieved results are shown and a comparison with another method is carried out to analyze the performance of our new approach.

**Keywords:** video activity recognition; common spatial patterns; histogram of optical flow

## 1. Introduction

Video Activity Recognition aims to automatically analyze and interpret particular events within video sequences. In the last years action recognition has gained interest, thanks to the growth of multimedia files availability and due to the amount of tasks it is useful for.

Currently, multimedia information generates large volumes of data, and this increases the need of developing automatic or semi-automatic systems which allow the labelling of video actions with different applications. For example, video cameras record data from different environments in real-time, which is useful, for instance, in security matters.

Many different domains can take advantage from video activity recognition, such as video security, video retrieval, or human-computer interaction. There are many different situations where a system that warns about suspicious actions in real time is highly beneficial, and since the enormous growth that multimedia data has experienced in recent years makes manual tagging tedious and sometimes impractical, video recognition can be used to perform annotation and indexing of videos. Furthermore, it can be useful to generate interactive systems for social purposes or for entertainment industry.

Identifying human actions in videos is a complex task. It is more challenging than image recognition, where a single frame represents the whole scene. In video recognition, a frame of a video where someone appears walking could also be a frame of a sequence of someone running, more frames are needed to see what action is taking place. The complexity of this task comes from the high intra-class variability that exists between the instances. This intra-class variability is caused by different factors, such as the diversity among people both in their appearance and in the style of execution of the action,

the movements of the camera, the environment, which is usually affected by changes in lighting, shadows or occlusions, the viewpoint and the distance of the subject from the camera, and other factors, such as differences in resolution. Human actions are associated with a spatial and a temporal component, both random, so the performance of the same action is never identical.

In this paper, taking as a basis the work of Reference [1], a new approach for video action recognition is presented, where Common Spatial Pattern (CSP) algorithm is used, a method which is normally used in Brain Computer Interface (BCI) for electroencephalography (EEG) systems [2]. CSP is a dimensionality reduction technique which consists of finding an optimum spatial filter to separate a multidimensional signal into two classes, maximizing the variance of one of them while minimizing the variance of the other. In our approach input videos are represented as frame sequences and the temporal sequence of each pixel is treated as a signal (channel) to feed the CSP. After CSP is applied, some signals descriptors are selected for classification purposes. In classical CSP applications, only the signal variances and Linear Discriminant Analysis (LDA) classifier [3] are used; in this research, variances, minimum, maximum, and interquartil range (IQR) are taken as descriptors, and LDA, K Nearest Neighbors (KNN) [4] and Random Forests (RF) [5] as classifiers.

The rest of the paper is organized as follows—First, in Section 2 some related works are mentioned in order to introduce the topic. In Section 3 a theoretical framework is presented to explain the proposed approach in detail. In Section 4 the experimental setup is presented, the used data-set and the different experimentation carried out are explained thoroughly. To conclude, in Section 5 the obtained results are shown and a comparison between our approach and another method is made.

## 2. Related Work

Different trends have been identified when it comes to video action recognition. Several approaches have been developed to deal with this problem along the years [6–8]. The existing techniques can be divided in four main groups: the identification of space-time interest points, the representation of action sequence as 3D spatio-temporal volume, the use of motion information and the use of deep learning to process sequences of frames.

Space-time interest points extracted from video have been widely used for action recognition. For instance, the authors of Reference [9] extract accumulated holistic features from clouds of interest points in order to use the global spatiotemporal distribution of interest points. This is followed by an automatic feature selection. Their model captures robust and smooth motions, where denser and more informative interest points are obtained. In Reference [10] a compact video representation is presented, using 3D Harris and 3D SIFT for feature extraction. K-means clustering is used to form a visual word codebook which is later classified by a Support Vector Machine (SVM) and a Naive Bayes classifiers. The authors of Reference [11] apply surround suppression together with local and temporal constraints to achieve a robust and selective STIP detection. A vocabulary of visual-words is built with a bag-of-video words (BoVW) model of local N-jet features and a Support Vector Machine (SVM) is used for classification.

In order to try to improve the activity recognition, RGB Depth (RGB-D) cameras are used (e.g., Microsoft Kinect, Intel RealSense), which are robust to illumination changes. The authors of Reference [12] extract random occupancy pattern (ROP) semi-local features from depth sequences captured by depth cameras. These features are encoded with a sparse coding approach. The training phase of the presented approach is fast, robust and it does not require careful parameter tuning. In Reference [13] both RGB and Depth Camera are used to extract motion features, generating a Salient Information Map. For each motion history image, a Complete Local Binary descriptor is computed, extracting sign, magnitude and center descriptors from the Salient Information Map. Canonical Correlation Analysis and dimensionality reduction are used to combine depth and RGB features. The classification is performed by a multiclass SVM.

Approaches which focus in motion information usually rely on optical flow or appearance. In Reference [14] the authors propose dense trajectories to describe videos. From each frame dense



trajectories are extracted and a dense optical flow algorithm is used to track them. To encode this information, the authors introduce a descriptor based on motion boundary histograms. They improve their work in Reference [15] by taking into account camera motion, using SURF descriptors and dense optical flow to match feature points between frames. A human detector is also used to avoid inconsistent matches between human motion and camera motion. The authors of Reference [16] decompose visual motion into dominant and residual motions. Then, they propose a new motion descriptor based on differential motion scalar quantities: divergence, curl and shear; the DCS descriptor, which captures additional information on the local motion patterns. The VLAD coding technique is used.

Recently, deep models have gained interest due to the good results they have obtained for image recognition. They are able to learn multiple layers of features hierarchies and automatically build high-level representations of the raw input. For video action recognition, Convolutional Neural Network (CNN) has been the most used model, extracting frames from videos and automatically classifying them by sending them as input features for the network. However, this way, temporal information is ignored and only spatial features are learnt. In Reference [17] they propose a two-stream CNN, where both spatial and temporal information are incorporated. The input of the spatial network is composed by the frames extracted from videos, whereas that of the temporal network is formed by the dense optical flow. Then, these two CNNs are combined by late fusion. Recurrent Neural Networks (RNNs) have also been proven to be effective for video activity recognition, specially Long Short-Term Memory (LSTM) networks. The authors of Reference [18] propose the use of a CNN along with a deep bidirectional LSTM (DB-LSTM) network. First, they use pre-trained AlexNet to extract deep features from every sixth frame of the videos. Then, sequence information from the features of video frames are learnt by the DB-LSTM. In Reference [19] the authors present a two-stream attention based LSTM network, which focuses on the effective features and assigns different weights to the outputs of each deep feature maps. They also propose a correlation network layer to identify the information loss and adjust the parameters.

### 3. CSP-Based Approach

Similar to Reference [1], the use of CSP is the main idea of the presented approach, although this time the signals to be processed are composed by temporal sequences of pixel. In this section, the used algorithm is explained, as well as the approach that is being introduced.

The Common Spatial Pattern (CSP) algorithm [20], a mathematical technique applied in signal processing, has been widely used in Brain Computer Interface (BCI) applications for electroencephalography (EEG) systems [21–23]. Research has also been published applying CSP in the field of electrocardiography (ECG) [24], electromyography (EMG) [25,26] or even in astronomical images for planet detection [27]. CSP was presented as an extension of Principal Component Analysis (PCA) and it consists of finding an optimum spatial filter which reduces the dimensionality of the original signals. Considering just two different classes, a CSP filter maximizes the difference of the variances between the classes, maximizing the variance of filtered signals of EEG of one of the targets while minimizing the variance for the other.

It must be clarified that, although the CSP algorithm has been used mainly with EEG problems, in this paper a new application is presented, the use of CSP filters for feature extraction in the human action recognition task. In our approach, each video represents a trial and each pixel is treated as an EEG channel, so the videos are taken as time series where the pixels are the channels which change over time. This is an application outside the usual field of analysis of physiological signals, somehow justified by the successful use in astronomical image processing [27], but here it is extended to videos depicting actions.

The full process can be seen in Figure 1. The first step consists of selecting the most relevant pixels of the frame sequences, that will be used to feed the CSP. In order to select the most relevant pixels, those which have the biggest variance are chosen, that is, the pixels that change most in the frame

sequence. Once the pixels are selected and, hence, the signals are formed, the CSP is computed in order to separate the classes according to their variance.

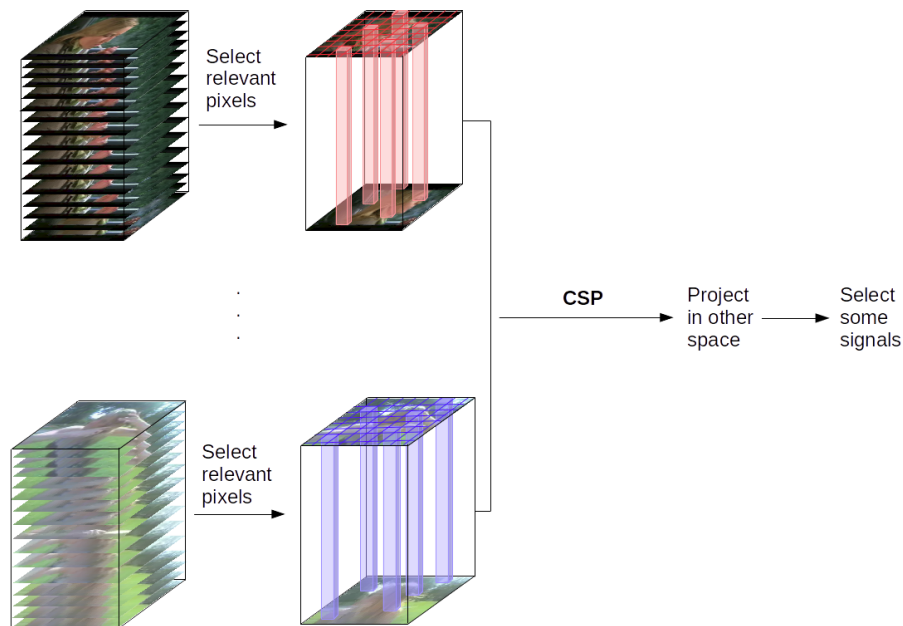


Figure 1. Proposed approach.

#### 4. Experimental Setup

In this section the details of the experiments that have been carried out are explained. First, the database used is presented. Then, different modalities are introduced and, finally, the optical flow method is explained, which has been used to make a comparison with the CSP approach. In Figure 2 a graphical overview of the presented approach is shown.

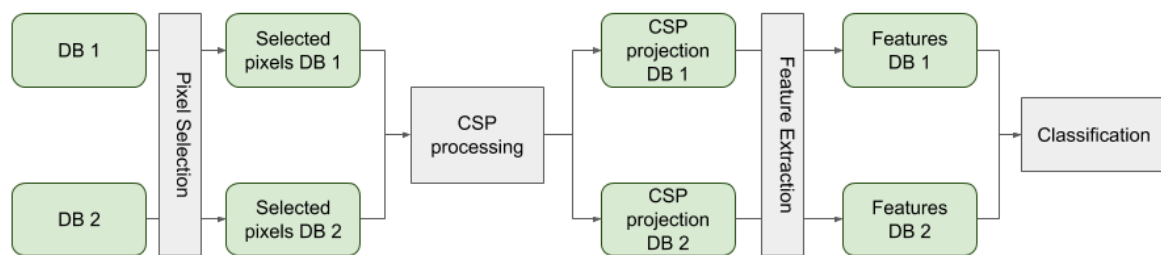


Figure 2. An overview of the full process of the presented technique.

HMDB51 <http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database> [28] is an action recognition database which collects videos from various sources, mainly from movies but also from public databases such as YouTube, Google and Prelinger Archives. It consists of 6849 videos with 51 action categories and a minimum of 101 clips belong to each category. The action categories can be divided into 5 main groups:

1. General facial actions: smiling, laughing, chewing, talking.
2. Facial actions with object manipulation: smoking, eating, drinking.
3. General body movements: cartwheeling, clapping hands, climbing, going up stairs, diving, falling down, backhand flipping, handstanding, jumping, pull-ups, push-ups, running, sitting down, sitting up, somersaulting, standing up, turning, walking, waving.
4. Body movements with object interaction: brushing hair, catching, drawing a sword, dribbling, playing golf, hitting something, kicking a ball, picking something, pouring, pushing something,

riding a bike, riding a horse, shooting a ball, shooting a bow, shooting a gun, swinging a baseball bat, drawing sword, throwing.

5. Body movements for human interaction: fencing, hugging, kicking someone, kissing, punching, shaking hands, sword fighting.

Apart from the action label, other meta-labels are indicated in each clip. Those labels provide information about some features describing properties of the clip, such as camera motion, lighting conditions, or background. As some videos are taken from movies or YouTube, the variation of features is high and that extra information can be useful. The quality of the videos has also been measured (*good, medium, bad*), and they are rated depending on whether body parts vanish while the action is executed or not. It is worth mentioning that this extra information has not been used in this paper.

For our experiments only 6 classes have been selected due to the large amount of images. The selected classes are *brushing hair, cartwheeling, fencing, punching, smoking* and *walking*. To work with videos, their frames have been extracted in the first place. It has been decided to extract the same number of frames in every video of each class, so the largest video has been selected and the number of frames of the videos of that class is defined by it (in order not to cut any video and maybe lose the action performance). In Table 1 the number of videos and number of frames are indicated for each class. In the case of HMDB51 as the number and length of the videos vary a lot, the number of frames changes in each class. However, the process to get the frames is the same, first the largest video is selected and it is used to determine the number of frames for the class. As some videos need to be extended to get the determined length, some of the frames of these videos are repeated.

**Table 1.** Data-set class details.

	Videos	Frames/Video	Frames/Class
Brush hair	107	648	69,336
Fencing	116	301	34,916
Walk	282	534	150,588
Punch	126	502	63,252
Smoke	109	445	48,505
Cartwheel	103	132	13,596

Due to the difference between the number of videos of some of the classes, it has been decided to use the same amount of videos for both classes when performing the classification. The class with fewer instances indicates the number of videos in each of the experiments. For instance, when performing the classification between *fencing* (116) and *smoke* (109) 109 videos are used from each class, with a total of 218 videos. Since the class with the fewest instances has 103 videos, it was decided that it is a sufficient amount of instances to do the tests without having to apply any other more complicated balancing method.

It must also be mentioned that in order to perform all the introduced experiments, the size of the images is set to  $25 \times 25$  due to the computational requirements of CSP. Moreover, the used CSP method is implemented to work with just 2 classes, therefore all the tests have been carried out using pairs of classes, one versus one [29].

#### 4.1. Experiments

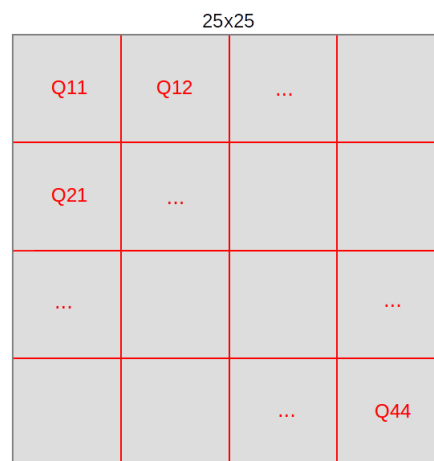
In this section the performed experiments are presented. They were developed using the technique introduced in Section 3, where the main idea is to consider the temporal sequence of pixel  $i, j$  as a signal to be fed to the CSP. Taking that algorithm as a basis, different methods have been computed to make a comparison between them and see which one performs better.

## Modalities

CSP algorithm was used in several modalities. The main change between the performed tests consists of deciding which pixels of the image are used to make the signals (channels) to calculate the CSP.

- Separation in quadrants.

Taking into account that different actions have to be recognized, where in the video they occur should be taken into account, that is, in which area of the window. In many of the sequences used, the camera is static and the individuals performing the interaction appear almost centered. In these cases, it can be supposed that if the action they are performing is, for instance, *smoke*, it will be happening at the top of the images, while if the action is *kick*, it will be happening at the bottom part. In order to consider this approach, it was decided to divide each frame of the video in 16 quadrants as can be seen in Figure 3, and perform the whole classification pipeline for every one of them. Thus, each classifier focuses on an exact area of the videos. The final prediction is the result of the majority voting of these 16 classifiers.



**Figure 3.** The division of a  $25 \times 25$  frame in 16 quadrants.

- Pixels with maximum variance.

CSP works with the variance of each channel. In some of the video sequences some objects from the scene, such as the background, are static, and do not change over time. The pixels corresponding to these static areas will produce a zero (or near zero, due to random fluctuations in pixel intensity) variance. This, apart from not providing useful information, can cause some problems at the time of executing the algorithm due to the calculation of the logarithm of the variances in the computation of CSP, yielding negative infinite values. In this case, it was decided to first select a group of pixels to extract the features, these are the pixels that change the most over time and therefore have the maximum variances. The hypothesis here is that they are the best candidates to represent the performed action. To select these pixels, the frames were transformed to grayscale, in order to have just one channel when calculating the variance of each pixel. In Figure 4 an example can be seen, where the 25 most relevant pixels have been selected for each quadrant.

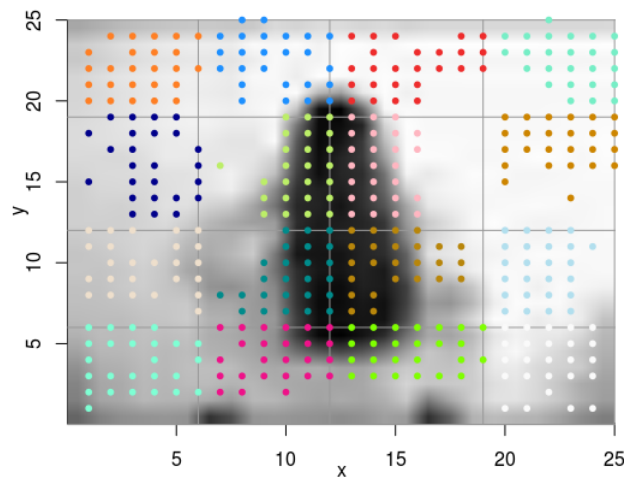


Figure 4. Selected pixels example.

Once the relevant pixels are selected and the quadrant separation has been decided, the classification is performed using different features extracted after the CSP filter. The main focus of the experimentation is the use of the variances of the signals after applying the Common Spatial Pattern filter. However, apart from the variances, many other information can be extracted from these transformed signals. Hence, some experiments are performed with just the information of the variances and other experiments also with information about the maximum and minimum values of the signal and the interquartile range ( $IQR = Q3 - Q1$ ). This information may be useful when performing the classification, and a comparison has been made between the results obtained by these two ways of performance:

- var
- var, min, max, IQR taken together

#### 4.2. Optical Flow

A comparison between the use of optical flow vectors and CSP features has been made, in order to analyze which features provide the best information about the action of the videos.

There are some algorithms which compute the optical flow. In this paper the OpenCV implementation of Gunnar Farneback's algorithm [30] is used. It provides a dense optical flow, which means that it calculates the optical flow for every point of the image. Dense techniques are slower but can be more accurate than sparse ones, which only compute the optical flow for some points of interest of the image. The result of Farneback's method is a two dimensional array containing the vectors which represent the displacement of each point of the scene.

After having calculated the optical flow for every pixel, the vectors are divided into 10 bins and, according to the gradient direction of each pixel, a histogram is created. To create the histogram, the directions are separated in 8 stripes as can be seen in Figure 5. The features that are then trained are taken from this information.

Once the features for each video have been obtained, the classification is performed.

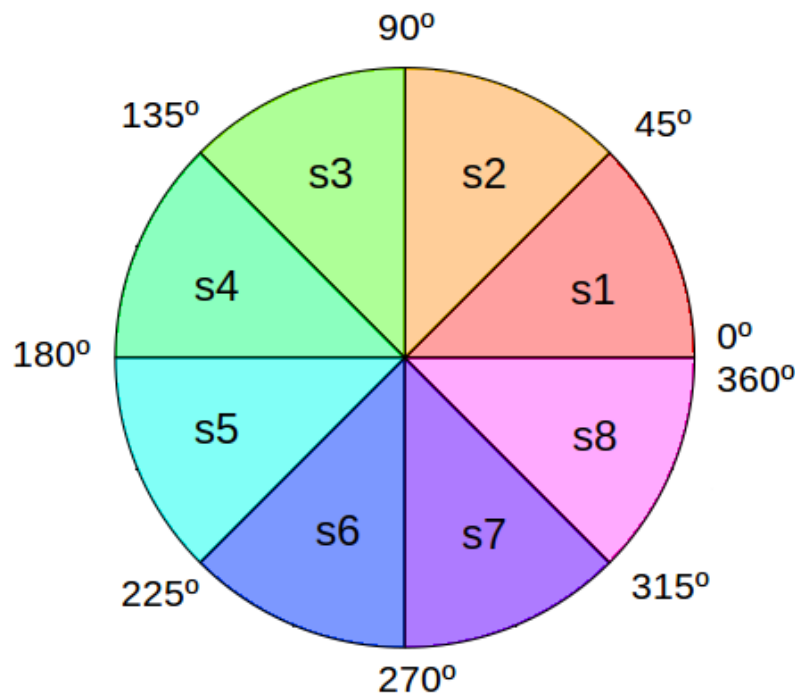


Figure 5. Direction of optical flow.

## 5. Experimental Results

After explaining how the process has been defined, the experimental results are presented. There are seven different accuracy results for each pair of classes:

1. HOF: the results obtained with the optical flow vectors as features.
2. Variance: after calculating the CSP, only the variances are taken as features.
  - (a)  $q = 3$  : 6 ( $2 \cdot q$ ) variance values are used.
  - (b)  $q = 5$  : 10 ( $2 \cdot q$ ) variance values are used.
  - (c)  $q = 10$  : 20 ( $2 \cdot q$ ) variance values are used.
3. More info: after calculating the CSP, apart from the variances, the minimum, the maximum and the IQR values of the curve are also taken as features.
  - (a)  $q = 3$  : 6 ( $2 \cdot q$ ) variance values are used, plus three additional features (min, max, IQR).
  - (b)  $q = 5$  : 10 ( $2 \cdot q$ ) variance values are used, plus three additional features (min, max, IQR).
  - (c)  $q = 10$  : 20 ( $2 \cdot q$ ) variance values are used, plus three additional features (min, max, IQR).

Variable  $q$  indicates how many feature vectors are considered in the projection, sorting the feature vectors of the spatial filter by variance the  $q$  first and  $q$  last vectors are selected. Therefore, a feature vector of  $2 \times q$  dimensionality is obtained after applying CSP. Exactly,  $q$  first and  $q$  last vectors are used, which yield the smallest variance for one class and simultaneously, the largest variance for the other class.

The results for classifiers KNN, LDA and RF can be seen in Tables 2–4, respectively. In each table the accuracy values obtained for every pair of the selected classes of HMDB51 database are presented, with the best values in bold. These results have been obtained by dividing the images in sixteen quadrants and getting the 25 pixels that change the most over time (the ones with the greater variance) for each quadrant, taking into account 400 pixels (channels) out of 625 ( $25 \times 25$ ).

Table 2. K Nearest Neighbors (KNN) classifier results.

KNN	Fencing	Walk	Punch	Smoke	Cartwheel	
Brush_hair	HOF: 0.7424	HOF: 0.7679	HOF: 0.8125	HOF: 0.5606	HOF: 0.8915	
	Variance: q = 3: 0.8730 q = 5: 0.8492 q = 10: 0.8492	Variance: q = 3: <b>0.7857</b> q = 5: 0.7381 q = 10: 0.7778	Variance: q = 3: 0.8016 q = 5: 0.8095 q = 10: 0.8095	Variance: q = 3: 0.7143 q = 5: <b>0.7460</b> q = 10: <b>0.7460</b>	Variance: q = 3: 0.9683 q = 5: 0.9603 q = 10: 0.9762	
	More info: q = 3: <b>0.8889</b> q = 5: 0.8809 q = 10: 0.8810	More info: q = 3: <b>0.7857</b> q = 5: 0.7540 q = 10: 0.7619	More info: q = 3: <b>0.8174</b> q = 5: 0.7778 q = 10: 0.7778	More info: q = 3: 0.7222 q = 5: 0.7301 q = 10: <b>0.7460</b>	More info: q = 3: <b>0.9921</b> q = 5: 0.9524 q = 10: 0.9365	
Fencing		HOF: 0.5865	HOF: 0.7778	HOF: 0.6212	HOF: 0.8605	
		Variance: q = 3: 0.6594 q = 5: <b>0.6812</b> q = 10: 0.6304	Variance: q = 3: <b>0.8043</b> q = 5: 0.6667 q = 10: 0.7246	Variance: q = 3: 0.6742 q = 5: 0.6364 q = 10: 0.5454	Variance: q = 3: 0.8254 q = 5: 0.8571 q = 10: 0.8571	
		More info: q = 3: 0.6739 q = 5: 0.6739 q = 10: 0.5797	More info: q = 3: 0.7681 q = 5: 0.7464 q = 10: 0.7681	More info: q = 3: <b>0.7273</b> q = 5: 0.6364 q = 10: 0.5303	More info: q = 3: <b>0.9127</b> q = 5: 0.7698 q = 10: 0.8333	
Walk			HOF: 0.6827	HOF: <b>0.7637</b>	HOF: <b>0.8248</b>	
			Variance: q = 3: 0.5800 q = 5: 0.6600 q = 10: 0.6067	Variance: q = 3: 0.5530 q = 5: 0.6061 q = 10: 0.5682	Variance: q = 3: 0.8174 q = 5: 0.7857 q = 10: 0.7222	
			More info: q = 3: 0.6333 q = 5: <b>0.6867</b> q = 10: 0.5800	More info: q = 3: 0.5303 q = 5: 0.6818 q = 10: 0.5000	More info: q = 3: 0.8016 q = 5: 0.8174 q = 10: 0.7778	
Punch				HOF: <b>0.6805</b>	HOF: <b>0.9078</b>	
				Variance: q = 3: 0.5682 q = 5: 0.5757 q = 10: 0.5303	Variance: q = 3: 0.8730 q = 5: 0.8651 q = 10: 0.8968	
				More info: q = 3: 0.6061 q = 5: 0.6364 q = 10: 0.5227	More info: q = 3: 0.8492 q = 5: 0.8730 q = 10: 0.8492	
Smoke					HOF: <b>0.8295</b>	
					Variance: q = 3: 0.7778 q = 5: 0.8016 q = 10: 0.7460	
					More info: q = 3: 0.8254 q = 5: 0.7619 q = 10: 0.7698	

**Table 3.** Linear Discriminant Analysis (LDA) classifier results.

LDA	Fencing	Walk	Punch	Smoke	Cartwheel
Brush_hair	HOF: 0.6591	HOF: 0.6920	HOF: 0.6180	HOF: 0.6288	HOF: 0.7752
	Variance: q = 3: 0.8492 q = 5: 0.7302 q = 10: 0.8809	Variance: q = 3: 0.7143 q = 5: 0.7540 q = 10: 0.7698	Variance: q = 3: <b>0.8333</b> q = 5: 0.8016 q = 10: 0.7699	Variance: q = 3: 0.7460 q = 5: 0.7778 q = 10: 0.6826	Variance: q = 3: 0.9365 q = 5: 0.9603 q = 10: 0.9365
	More info: q = 3: 0.9445 q = 5: <b>0.9683</b> q = 10: <b>0.9683</b>	More info: q = 3: 0.6984 q = 5: 0.7540 q = 10: <b>0.7857</b>	More info: q = 3: 0.8174 q = 5: 0.7936 q = 10: 0.8016	More info: q = 3: 0.7699 q = 5: 0.7936 q = 10: <b>0.8254</b>	More info: q = 3: <b>0.9921</b> q = 5: <b>0.9921</b> q = 10: <b>0.9921</b>
Fencing		HOF: 0.6414	HOF: 0.7083	HOF: 0.7803	HOF: 0.7984
		Variance: q = 3: 0.6522 q = 5: 0.6377 q = 10: 0.6739	Variance: q = 3: 0.7174 q = 5: 0.6957 q = 10: 0.7246	Variance: q = 3: 0.6818 q = 5: 0.6667 q = 10: 0.6212	Variance: q = 3: 0.8571 q = 5: 0.8651 q = 10: 0.9048
		More info: q = 3: <b>0.8188</b> q = 5: 0.7826 q = 10: <b>0.8188</b>	More info: q = 3: 0.8768 q = 5: <b>0.9058</b> q = 10: 0.8406	More info: q = 3: 0.8030 q = 5: 0.7651 q = 10: <b>0.8182</b>	More info: q = 3: <b>0.9921</b> q = 5: 0.9286 q = 10: 0.9603
Walk			HOF: 0.6546	HOF: 0.6287	HOF: 0.8889
			Variance: q = 3: 0.6000 q = 5: 0.6267 q = 10: 0.6067	Variance: q = 3: 0.5379 q = 5: 0.5303 q = 10: 0.6061	Variance: q = 3: 0.7381 q = 5: 0.7778 q = 10: 0.7460
			More info: q = 3: <b>0.7000</b> q = 5: 0.6667 q = 10: 0.5600	More info: q = 3: 0.6136 q = 5: <b>0.7046</b> q = 10: 0.6515	More info: q = 3: 0.9524 q = 5: 0.9524 q = 10: <b>0.9603</b>
Punch				HOF: <b>0.7847</b>	HOF: 0.9007
				Variance: q = 3: 0.5757 q = 5: 0.6667 q = 10: 0.5076	Variance: q = 3: 0.8969 q = 5: 0.9127 q = 10: 0.8413
				More info: q = 3: 0.7046 q = 5: 0.7197 q = 10: 0.7273	More info: q = 3: 0.9841 q = 5: <b>0.9921</b> q = 10: 0.9445
Smoke					HOF: 0.7985
					Variance: q = 3: 0.7778 q = 5: 0.7698 q = 10: 0.8412
					More info: q = 3: <b>0.9841</b> q = 5: 0.9762 q = 10: 0.9683



Table 4. RF classifier results.

RF	Fencing	Walk	Punch	Smoke	Cartwheel
Brush_hair	HOF: 0.7576	HOF: <b>0.8607</b>	HOF: <b>0.8680</b>	HOF: 0.6894	HOF: 0.9535
	Variance: q = 3: 0.8413 q = 5: 0.8730 q = 10: 0.8651	Variance: q = 3: 0.7540 q = 5: 0.6984 q = 10: 0.7698	Variance: q = 3: 0.8095 q = 5: 0.8254 q = 10: 0.8254	Variance: q = 3: 0.7143 q = 5: 0.6587 q = 10: 0.7302	Variance: q = 3: 0.9286 q = 5: 0.9445 q = 10: 0.9445
	More info: q = 3: <b>0.9683</b> q = 5: 0.9603 q = 10: 0.9445	More info: q = 3: 0.7460 q = 5: 0.7222 q = 10: 0.8174	More info: q = 3: 0.8492 q = 5: 0.8016 q = 10: 0.8254	More info: q = 3: <b>0.8016</b> q = 5: 0.7619 q = 10: 0.7857	More info: q = 3: <b>0.9841</b> q = 5: 0.9762 q = 10: 0.9762
Fencing		HOF: 0.7553	HOF: 0.8889	HOF: 0.8106	HOF: 0.9380
		Variance: q = 3: 0.7391 q = 5: 0.7391 q = 10: 0.7391	Variance: q = 3: 0.7681 q = 5: 0.7464 q = 10: 0.7391	Variance: q = 3: 0.7576 q = 5: 0.7348 q = 10: 0.8030	Variance: q = 3: 0.8571 q = 5: 0.8016 q = 10: 0.8889
		More info: q = 3: 0.9058 q = 5: 0.8913 q = 10: <b>0.9130</b>	More info: q = 3: 0.8406 q = 5: 0.8696 q = 10: <b>0.9058</b>	More info: q = 3: 0.8561 q = 5: 0.8561 q = 10: <b>0.8712</b>	More info: q = 3: <b>0.9445</b> q = 5: 0.8889 q = 10: 0.8412
Walk			HOF: <b>0.7912</b>	HOF: <b>0.7384</b>	HOF: 0.9103
			Variance: q = 3: 0.6800 q = 5: 0.6667 q = 10: 0.6000	Variance: q = 3: 0.4924 q = 5: 0.6288 q = 10: 0.5909	Variance: q = 3: 0.8413 q = 5: 0.8730 q = 10: 0.7937
			More info: q = 3: 0.6467 q = 5: 0.6733 q = 10: 0.6667	More info: q = 3: 0.6212 q = 5: 0.6439 q = 10: 0.5758	More info: q = 3: <b>0.9841</b> q = 5: 0.9683 q = 10: 0.9445
Punch				HOF: <b>0.8403</b>	HOF: 0.9504
				Variance: q = 3: 0.6742 q = 5: 0.7500 q = 10: 0.7348	Variance: q = 3: 0.8651 q = 5: 0.8968 q = 10: 0.8492
				More info: q = 3: 0.6894 q = 5: 0.7500 q = 10: 0.7424	More info: q = 3: <b>0.9841</b> q = 5: 0.9762 q = 10: <b>0.9841</b>
Smoke					HOF: 0.8992
					Variance: q = 3: 0.8095 q = 5: 0.9048 q = 10: 0.8730
					More info: q = 3: 0.9445 q = 5: <b>0.9762</b> q = 10: <b>0.9762</b>

Analyzing the results, it can be seen that the KNN classifier obtains the worst results. The other two classifiers, LDA and RF, achieve more similar results, although results with RF classifier are, in general, better. That is, for most pairs of classes, the KNN classifier gets the lowest accuracy values. The best accuracy for each pair is sometimes achieved by LDA and other times by RF, RF being the one which gets the best outcomes most of the times.

Regarding the feature extraction techniques, there is not a clear winner. The best results are highlighted in the mentioned tables. As it can be seen, depending on the targets and the classifiers, one technique or the other is preferred. There is not much difference between the use of the different features. As the obtained results are not clear enough to determine which features are better to use, a statistical test is performed to compare them.

Before performing the comparison, another aspect of the classification must be mentioned, the target classes. Depending on the classes that are being classified, the results may vary, because some pairs can be more distinguishable than others. For instance, the pair *brush hair* and *cartwheel* classes get high accuracy values for every technique and algorithm, with a mean of 0.95. However, other classes such as *walk* are more difficult to discriminate, no matter what algorithm, technique or even what class it is compared with. This could be due to the videos related to the class, since they can be confusing and the information may not be well represented. It must also be mentioned that when the resolution of the images is  $25 \times 25$  the obtained results are surprisingly good.

In Table 5 a summary of the results is presented. For each classifier, the results are divided in *variance*, *more info* and *HOF*. The value of *best* is calculated with the mean of the best values of every pair of classes. However, the *mean best* value is the mean of the results of the configuration which gets the best mean result calculated with every pair of classes. The *best* value is an optimistic summary, due to the fact that only the best values are taken into account, while the value presented in *mean best* is more realistic. As the *HOF* method only has one value per pair of classes (because it does not have the *q* parameter), the mean between these values is the value indicated in the table. Looking at the results, it can be seen that RF classifier gets the best values. Besides, when more information is used, better values are obtained.

It can be observed that when just variance is used, LDA gets the worst results and, furthermore, when more information is used, KNN is the worst with a remarkable difference. Regarding the methods, it can be observed that *more info* and *HOF* methods get more similar results than *variance* methods, at least for some of the classifiers.

**Table 5.** Summary results for each classifier.

	Variance		More Info		HOF
	Best	Mean Best	Best	Mean Best	
KNN	0.7572	0.7355	0.7752	0.7503	0.754
LDA	0.7571	0.7273	<b>0.85</b>	0.8298	0.7305
RF	<b>0.7902</b>	<b>0.7646</b>	<b>0.8567</b>	<b>0.8365</b>	<b>0.8435</b>

### Comparison

Deciding whether our approach is better than the Histogram of Optical Flow is not trivial and can not be assumed by the obtained results due to the lack of clear differences. For that reason, a statistical test needs to be performed to determine if there is a difference between the approaches presented and if the test indicates that a difference exists, we could determine which model is better by the individual results.

The statistical test that has been used is the *Friedman Test*. The Friedman Test is used when the data is dependant and it can be considered as an extension of Wilcoxon signed-rank test for more than two groups. The Friedman Test is computed this way:

1. Being  $\{x_{ij}\}_{m \times n}$  a data table with  $m$  rows and  $n$  columns,  $\{r_{ij}\}_{m \times n}$  is calculated where  $r_{ij}$  is the order of  $x_{ij}$  in every block  $i$ .
2. Then, the statistic is calculated:

$$Q = \frac{12m}{n(n+1)} \sum_{j=1}^n \left( \bar{r}_j - \frac{k+1}{2} \right)^2 \tag{1}$$

$$\bar{r}_j = \frac{1}{m} \sum_{i=1}^m r_{ij}. \tag{2}$$

3. Finally, the p-value is defined this way, approximating the probability distribution of  $Q$  by a  $\chi^2$  distribution:

$$P(\chi_{n-1}^2 \geq Q). \tag{3}$$

When applying this test, the null hypothesis is that there are no differences between the tested groups. As in Equation (3), if the calculated probability is lower than the significance level, the null hypothesis is rejected, which indicates that at least 2 of the tested groups are significantly different from each other. In our approach, the hypotheses are defined this way:

- H0: there is no difference between the tested models  $\Leftrightarrow p\text{-value} \geq 0.05$
- H1: at least 2 of the tested models are different from each other  $\Leftrightarrow p\text{-value} < 0.05$

The dependant variable is formed by the accuracy values obtained with each model, the grouping variable is the definition of the models and the blocking variable the ranking of the models, from 1 to 15 in our case.

After computing the Friedman Test, the obtained results indicate that there is evidence that at least two of the tested models are different ( $\chi^2(20) = 123.68, p\text{-value} < 2.2 \times 10^{-16}$ ), therefore the null hypothesis is rejected. Although it has been proven that a difference exists between the tested models, a post-hoc analysis is required to determine which groups are significantly different from each other. To do so, we have used the Nemenyi test [31].

In Table 6 the obtained  $p$ -values can be seen. As our objective is to compare our results to the ones achieved by the Histogram of Optical Flow method, the values that are presented in Table 6 are specifically these comparisons. The names of our approaches are defined by the regular expression of Equation (4).

$$CSP(3|5|10)(\_var)? - ((KNN)|(LDA)|(RF)) \tag{4}$$

where 3, 5, 10 indicate the  $q$  value,  $\_var$  indicates if just the variance is taken as feature and KNN, LDA or RF define the algorithm that is used to create the model.

In the obtained results most of the  $p$ -value are not significant. However, there are some of them that indicate an evident difference between the models. The values in green are significant ( $<0.05$ ) and the values in red are very significant ( $<0.01$ ). There are a total of eleven pairs where a significant differences have been detected.

Referring to the original values, we observe that HOF-RF beats the CSP approach five times, while CSP-RF beats HOF-LDA three times.

It is not surprising to obtain better results with HOF-RF model, because the Random Forest classifier achieves better results than Linear Discriminant Analysis or K-Nearest Neighbors algorithms. Thus, in this case the difference and the adequacy of the Histogram of Optical Flow models is more related to the selected algorithm than to the features that are used to train the models.

As in the previous explanation, the CSP-RF against HOF-LDA results can be related to the selected algorithm without taking into account which features are used to train the models.

- CSP3-LDA and HOF-LDA,  $p\text{-value} = 0.1115$ .
- CSP5-LDA and HOF-LDA,  $p\text{-value} = 0.01053$ .

- CSP10-LDA and HOF-LDA,  $p$ -value = 0.01478.

**Table 6.** Fridman Nemenyi post-hoc results.

	HOF-KNN	HOF-LDA	HOF-RF
CSP3_var-KNN	1.00000	1.00000	0.10176
CSP3_var-LDA	0.99995	1.00000	<b>0.02410</b>
CSP3_var-RF	1.00000	1.00000	0.23000
CSP5_var-KNN	1.00000	1.00000	0.08550
CSP5_var-LDA	1.00000	1.00000	<b>0.04669</b>
CSP5_var-RF	1.00000	0.99417	0.79919
CSP10_var-KNN	0.99995	1.00000	<b>0.02410</b>
CSP10_var-LDA	0.99980	1.00000	<b>0.01651</b>
CSP10_var-RF	1.00000	0.99594	0.77079
CSP3-KNN	1.00000	0.99345	0.80825
CSP3-LDA	0.33847	<b>0.01115</b>	1.00000
CSP3-RF	0.23807	<b>0.00584</b>	1.00000
CSP5-KNN	1.00000	1.00000	0.19239
CSP5-LDA	0.32851	<b>0.01053</b>	1.00000
CSP5-RF	0.50285	<b>0.02541</b>	1.00000
CSP10-KNN	0.95234	1.00000	<b>0.00104</b>
CSP10-LDA	0.39047	<b>0.01478</b>	1.00000
CSP10-RF	0.25475	<b>0.00659</b>	1.00000

However, in the above three comparisons the same algorithm is used, Linear Discriminant Analysis, so in these cases, the significant differences are due to the selected features. Analyzing the performance of these models, it can be seen that CSP models beat HOF models, the features extracted by CSP are better for LDA classification than HOF features. In conclusion, our approach gets better results than Histogram of Optical Flow approach for at least one classification algorithm.

The rest of the comparisons do not show significant differences and they are considered as the same. However, our approach achieves better outcomes for some cases and maybe by carrying out different tests, the results would improve.

## 6. Conclusions

In this paper, a new approach for human activity recognition task is presented. It consists of the application of CSP (normally used in EEG systems) as feature extraction method before performing the classification. The resolution of the used videos is low (images of  $25 \times 25$ ) in order to complete different tests. After getting the results, HOF features are extracted and new models are created to compare them to the results obtained by the CSP method.

As previously mentioned, our approach gets good results taking into account the resolution of the images. Furthermore, after performing the statistical test, it has been shown that, at least for one of the methods, CSP features are better than HOF features.

As future work other databases could be used, or the extra information provided by HMDB51 data-set could be taken into account, since apart from the action label, other meta-labels are also indicated in each clip. This information could be useful to make some preprocessing before the feature extraction method. Besides, the resolution of the images could be improved if more computational capacity was obtained, doing tests with different sizes of images ( $25 \times 25$ ,  $40 \times 40$ ,  $60 \times 60$ , ...). In our approach three different classifiers have been used, but in the future more classifiers or even deep learning techniques could be applied after doing the feature extraction with the CSP method.

In conclusion, it is shown that with a simple method normally used for other tasks acceptable results can be obtained, without having to use very complicated ideas to achieve our goals.

**Author Contributions:** Research concept and supervision of technical writing: B.S. and I.I.; Software implementation, concept development and technical writing: I.R.-M.; Results validation and supervision of technical writing: J.M.M.-O. and I.G.; Methodological analysis: I.R.-R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially funded by the Basque Government, Research Teams grant number IT900-16, ELKARTEK 3KIA project KK-2020/00049, and the Spanish Ministry of Science (MCIU), the State Research Agency (AEI), and the European Regional Development Fund (FEDER), grant number RTI2018-093337-B-I100 (MCIU/AEI/FEDER, UE). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Rodríguez-Moreno, I.; Martínez-Otzeta, J.M.; Goienetxea, I.; Rodríguez-Rodríguez, I.; Sierra, B. Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns. *Sensors* **2020**, *20*, 2436. [[CrossRef](#)] [[PubMed](#)]
- Astigarraga, A.; Arruti, A.; Muguierza, J.; Santana, R.; Martín, J.I.; Sierra, B. User adapted motor-imaginary brain-computer interface by means of EEG channel selection based on estimation of distributed algorithms. *Math. Prob. Eng.* **2016**, *2016*, 1435321. [[CrossRef](#)]
- Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [[CrossRef](#)]
- Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
- Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.
- Ke, S.R.; Thuc, H.L.U.; Lee, Y.J.; Hwang, J.N.; Yoo, J.H.; Choi, K.H. A review on video-based human activity recognition. *Computers* **2013**, *2*, 88–131. [[CrossRef](#)]
- Rodríguez-Moreno, I.; Martínez-Otzeta, J.M.; Sierra, B.; Rodríguez, I.; Jauregi, E. Video activity recognition: State-of-the-art. *Sensors* **2019**, *19*, 3160. [[CrossRef](#)] [[PubMed](#)]
- Aggarwal, J.K.; Xia, L. Human activity recognition from 3d data: A review. *Pattern Recognit. Lett.* **2014**, *48*, 70–80. [[CrossRef](#)]
- Bregonzio, M.; Gong, S.; Xiang, T. Recognising action as clouds of space-time interest points. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1948–1955.
- Nazir, S.; Yousaf, M.H.; Velastin, S.A. Evaluating a bag-of-visual features approach using spatio-temporal features for action recognition. *Comput. Electr. Eng.* **2018**, *72*, 660–669. [[CrossRef](#)]
- Chakraborty, B.; Holte, M.B.; Moeslund, T.B.; González, J. Selective spatio-temporal interest points. *Comput. Vis. Image Underst.* **2012**, *116*, 396–410. [[CrossRef](#)]
- Wang, J.; Liu, Z.; Chorowski, J.; Chen, Z.; Wu, Y. Robust 3d action recognition with random occupancy patterns. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 872–885.
- Arivazhagan, S.; Shebiah, R.N.; Harini, R.; Swetha, S. Human action recognition from RGB-D data using complete local binary pattern. *Cogn. Syst. Res.* **2019**, *58*, 94–104. [[CrossRef](#)]
- Wang, H.; Kläser, A.; Schmid, C.; Liu, C.L. Action recognition by dense trajectories. In Proceedings of the CVPR 2011, Providence, RI, USA, 20–25 June 2011; pp. 3169–3176.
- Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 3551–3558.
- Jain, M.; Jegou, H.; Bouthemy, P. Better exploiting motion for better action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2555–2562.
- Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*; ACM: New York, NY, USA, 2014; pp. 568–576.
- Ullah, A.; Ahmad, J.; Muhammad, K.; Sajjad, M.; Baik, S.W. Action recognition in video sequences using deep bi-directional LSTM with CNN features. *IEEE Access* **2017**, *6*, 1155–1166. [[CrossRef](#)]
- Dai, C.; Liu, X.; Lai, J. Human action recognition using two-stream attention based LSTM networks. *Appl. Soft Comput.* **2020**, *86*, 105820. [[CrossRef](#)]

20. Fukunaga, K.; Koontz, W.L. Application of the Karhunen-Loève Expansion to Feature Selection and Ordering. *IEEE Trans. Comput.* **1970**, *C-99*, 311–318.
21. Ramoser, H.; Muller-Gerking, J.; Pfurtscheller, G. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 441–446. [[CrossRef](#)] [[PubMed](#)]
22. Wang, Y.; Gao, S.; Gao, X. Common spatial pattern method for channel selection in motor imagery based brain-computer interface. In Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 17–18 January 2006; pp. 5392–5395.
23. Novi, Q.; Guan, C.; Dat, T.H.; Xue, P. Sub-band common spatial pattern (SBCSP) for brain-computer interface. In Proceedings of the 2007 3rd International IEEE/EMBS Conference on Neural Engineering, Kohala Coast, HI, USA, 2–5 May 2007; pp. 204–207.
24. Alotaiby, T.N.; Alshebeili, S.A.; Aljafar, L.M.; Alsabhan, W.M. ECG-based subject identification using common spatial pattern and SVM. *J. Sens.* **2019**, *2019*, 8934905. [[CrossRef](#)]
25. Kim, P.; Kim, K.S.; Kim, S. Using common spatial pattern algorithm for unsupervised real-time estimation of fingertip forces from sEMG signals. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 5039–5045.
26. Li, X.; Fang, P.; Tian, L.; Li, G. Increasing the robustness against force variation in EMG motion classification by common spatial patterns. In Proceedings of the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo, Korea, 11–15 July 2017; pp. 406–409.
27. Shapiro, J.; Savransky, D.; Ruffio, J.B.; Ranganathan, N.; Macintosh, B. Detecting Planets from Direct-imaging Observations Using Common Spatial Pattern Filtering. *Astron. J.* **2019**, *158*, 125. [[CrossRef](#)]
28. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.
29. Mendialdua, I.; Martínez-Otzeta, J.M.; Rodríguez-Rodríguez, I.; Ruiz-Vázquez, T.; Sierra, B. Dynamic selection of the best base classifier in one versus one. *Knowl. Based Syst.* **2015**, *85*, 298–306. [[CrossRef](#)]
30. Farnebäck, G. Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conference on Image Analysis*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 363–370.
31. Nemenyi, P. Distribution-free multiple comparisons (Doctoral Dissertation, Princeton University, 1963). *Diss. Abstr. Int.* **1963**, *25*, 1233.

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

# A New Approach for Video Action Recognition: CSP-Based Filtering for Video to Image Transformation

<b>Title:</b>	A New Approach for Video Action Recognition: CSP-Based Filtering for Video to Image Transformation
<b>Authors:</b>	I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, I. Rodriguez, B. Sierra
<b>Journal:</b>	IEEE Access
<b>Publisher:</b>	IEEE
<b>DOI:</b>	10.1109/ACCESS.2021.3118829
<b>Year:</b>	2021
<b>Times cited:</b>	1 (Google Scholar)
<b>Source of impact:</b>	WOS (JCR)
<b>Category:</b>	COMPUTER SCIENCE, INFORMATION SYSTEMS
<b>Impact index:</b>	3.476 (Q2)
<b>Position:</b>	79/164





Received September 7, 2021, accepted October 2, 2021, date of publication October 8, 2021, date of current version October 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3118829

# A New Approach for Video Action Recognition: CSP-Based Filtering for Video to Image Transformation

ITSASO RODRÍGUEZ-MORENO<sup>1</sup>, JOSÉ MARÍA MARTÍNEZ-OTZETA<sup>1</sup>, IZARO GOIENETXEA<sup>1</sup>,  
IGOR RODRIGUEZ<sup>1</sup>, AND BASILIO SIERRA<sup>1</sup>

Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), 20018 Donostia-San Sebastián, Spain

Corresponding author: Itsaso Rodríguez-Moreno (itsaso.rodriguez@ehu.eus)

This work was supported in part by the Basque Government Research Teams under Grant IT900-16, in part by ELKARTEK 3KIA Project under Grant KK-2020/00049, in part by the Spanish Ministry of Science (MCIU), the State Research Agency (AEI), and the European Regional Development Fund (FEDER) (MCIU/AEI/FEDER, European Union (EU)) under Grant RTI2018-093337-B-I100, and in part by the Spanish Ministry of Science, Innovation and Universities under Grant FPU18/04737 (predoctoral grant).

**ABSTRACT** In this paper we report on the design of a pipeline involving Common Spatial Patterns (CSP), a signal processing approach commonly used in the field of electroencephalography (EEG), matrix representation of features and image classification to categorize videos taken by a humanoid robot. The ultimate goal is to endow the robot with action recognition capabilities for a more natural social interaction. Summarizing, we apply the CSP algorithm to a set of signals obtained for each video by extracting skeleton joints of the person performing the action. From the transformed signals a summary image is obtained for each video, and these images are then classified using two different approaches; global visual descriptors and convolutional neural networks. The presented approach has been tested on two data sets that represent two scenarios with common characteristics. The first one is a data set with 46 individuals performing 6 different actions. In order to create the group of signals of each video, OpenPose has been used to extract the skeleton joints of the person performing the actions. The second data set is an Argentinian Sign Language data set (LSA64) from which the signs performed using just the right hand have been used. In this case the joint signals have been obtained using MediaPipe. The results obtained with the presented method have been compared with a Long Short-Term Memory (LSTM) method, achieving promising results.

**INDEX TERMS** Action recognition, social robotics, global visual descriptors, common spatial patterns, sign language recognition.

## I. INTRODUCTION

Video action recognition is a task which involves recognizing the action that is being performed in a sequence of observations. It is mainly used in computer vision, since the visual features provide basic information about what is happening in the image sequence, and has many real-life applications, such as visual surveillance, rehabilitation, human-computer interaction or entertainment.

Due to the fast growth of the technology, the demand for automatic interpretation of human behavior within videos is also growing, making video action recognition a highly active area. Even though many different approaches have been pre-

sented throughout the years trying to solve the problem of the identification of actions in videos, action recognition has not seen the gains in performance that have been achieved in image classification or human face recognition. The main reason is the complexity of combining both spatial and temporal information, which makes this problem harder than image analysis.

In this paper, a pipeline for a video action recognition method is presented, which has been applied to solve two problems with common characteristics. The first application where the presented method has been tested is human-robot interaction. Human-robot interaction (HRI) aims to understand, design and evaluate robotic systems to be used by or with humans. Specially when dealing with social robots, a highly evolved type of interaction is required, since these

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasa<sup>1</sup>.



(a) Image captured by the robot. (b) Expected reaction of the robot.

**FIGURE 1. Interaction example.**

robots cannot be merely teleoperated, and they are expected to meet high operational standards in order to be accepted by the general public.

The presented method aims to endow a pseudo-humanoid robot with the ability to understand the action that an actor is performing, in order to be able to give an adequate response, thus enhancing the social capabilities of the robot. A data set with six different actions performed by different people has been created to test the method. In Fig. 1 an interaction example between a person and the robot is displayed.

The second application is the sign language recognition. Nowadays, a large number of people has some degree of hearing impairments, about 466 million, and this number is expected to grow in the next years. Many of those people use sign languages to communicate with others, but since these languages are not commonly known among the hearing community, people with hearing problems often face communication difficulties in environments where no interpreter is available. In order to try to break the barrier between the hearing impaired community and the rest of the society, significant work is being carried out in Sign Language Recognition (SLR), where computer vision is playing a major role.

In order to improve the interaction between the people with hearing impairments and the robot, it is interesting to endow the robot with the ability to recognize certain gestures and react in different ways. Driven by the results obtained in [1], it has been decided to test the method presented in this paper on the recognition of some signs that are included in an Argentinian Sign Language database.

The approach presented herein continues with the work presented in [2], where Common Spatial Patterns (CSP), a method commonly used in Brain Computer Interface (BCI) for ElectroEncephaloGram (EEG) systems [3], [4], is used as feature extraction method for a video action recognition task.

In order to apply CSP, the information about the person performing the action to identify must be extracted. To that end, two different technologies have been used: OpenPose [5] to extract the skeletons of the action recognition videos and MediaPipe [6] to extract hand landmarks of the sign language data set.

The positions of the joints of the skeletons are used as input for the CSP, as presented in the previous work [2]. In this new approach, after computing the CSP algorithm, a matrix multiplication is applied and the transformed signals are represented as images. The features for the classification are extracted from those images using several visual global descriptors, and different classifiers have been tested to perform the classification.

Several experiments have been performed with the proposed approach in both databases, and their results have been compared to a Long Short-Term Memory (LSTM) paradigm in order to validate it.

The rest of the paper is organized as follows. First, in Section II some related works are described in order to introduce the topic. In Section III the proposed approach is introduced, explaining the process that has been carried out. Then, in Section IV the databases are presented and the experiments are explained further. Next, in Section V the obtained results are shown, and finally, in Section VI the conclusions extracted from this work are presented and future work is pointed out.

## II. RELATED WORKS

Many approaches for video action recognition have been introduced lately. These techniques make use of the visual features extracted from the video, both static and temporal. The temporal features mix the static image features with time information, so that the temporal information of the video is maintained.

In [7] the authors use a temporal template which is based on a static vector-image where the value of the vector at each point represents a function of the motion properties at the corresponding spatial location in an image sequence. Local spatio-temporal interest points can be used to recognize complex motion patterns as it is demonstrated in [8]. A hybrid hierarchical model is presented in [9] where collections of spatial and spatio-temporal features are used to represent video sequences. Many other methods make use of Histograms of Oriented Gradients (HOG) or Histogram of Oriented Optical Flow (HOOF) [10]–[12]. Motion descriptors based on the direction of optical flow have also been introduced [13], [14]. The use of depth data captured by depth cameras has also grown due to the advances in imaging technology [15], [16].

With these two publications [17], [18] as a starting point, deep learning has continued to be used for activity recognition, mainly with Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) [19]. Very deep two-stream ConvNets are presented in [20] which, according to the authors, get close to image domain deep models. Convolutional Neural Networks (CNN) and deep bidirectional LSTM (DB-LSTM) networks are used in [21]. In [22] the authors combine 3D-CNN and LSTM networks. Motion maps, which integrate temporal information, are iteratively extracted from videos using a kind of deep 3-dimensional

CNN (C3D), acquiring a final motion map of the whole video. LSTM is used for the final prediction.

As two-stream CNNs are unable to model long-term temporal structures, Wang *et al.* [23] developed a temporal segment network (TSN) which is able to model dynamics throughout the whole video. TSN extracts short snippets over a long video sequence with a sparse sampling scheme, this way modeling long-range temporal structures and preserving relevant information. Temporal Relation Network (TRN) [24] is a network module which enables temporal relational reasoning and can be easily plugged into an existing neural network. The module tries to describe the temporal relations between observations in videos. While TSN uses average pooling ignoring the temporal order, TRN replaces the average pooling with an interpretable relational module. Authors of [25] proposed a Temporal Shift Module (TSM) which shifts the channels forward or backward along the temporal dimension to exchange information between adjacent frames. The Gate Shift Module (GSM) [26] has a learnable spatial gating block which controls spatio-temporal interactions. Other authors [27] present Channel-Separated Convolutional Networks (CSN), which factorize 3D convolutions in point-wise  $1 \times 1 \times 1$  convolutions for channel interaction or depth-wise  $k \times k \times k$  (usually  $k = 3$ ) convolutional operations for local spatio-temporal interactions. Temporal Pyramid Network (TPN) [28] models the visual tempo at feature level, extracting temporal features by combining features obtained at different tempos.

Skeleton data has also been used to perform activity recognition. The authors of [29] use a LSTM network to focus on the significant joints of the skeleton within each frame and, according to that, the outputs of different frames are weighted. In [30] the authors present a representation where a human pose estimator is used and heatmaps are extracted for the human joints in each frame. In [31] a method for encoding geometric relational features into color texture images is presented, where temporal variations of different features are converted into the color variations of their corresponding images. They use a multi-stream CNN model to classify the images. The authors of [32] propose a two-stream adaptive graph convolutional network (2s-AGCN), where both the coordinates of the joints and the bones between the joints are used as features for classification.

Regarding Sign Language Recognition (SLR), different techniques have been used in recent years [33]–[36]. On the one hand, we can find methods that make use of intrusive sensors which must be placed on the person who is performing the signs. These wearable markers or data gloves are used to detect the body and hand movements [37]–[39]. In the case of non-intrusive systems, there are techniques that make use of sensors such as Leap Motion or Microsoft Kinect [40]–[43] and others that focus on the information obtained by cameras, vision-based methods [44]–[46]. Most of the presented methods use neural networks to perform the classification, like CNNs and LSTMs [47]–[49], although Hidden Markov Models (HMM) have been widely used for

SLR too [50]–[52]. As a practical application, it is possible to mention [53], where the authors develop a software system for hearing impaired children with articulation disorders.

### III. PROPOSED APPROACH

The method presented in this paper is a continuation of the work presented in [2], which uses CSP applied to skeleton information for video action recognition. In this work, an image is obtained for each video that summarizes the information of the video and that can be then classified using image classifiers. Therefore, we transform the video classification problem into an image classification problem. An overview of the proposed approach can be seen in Fig. 2.

As seen in the overview of the method, the first step is the extraction of the skeletons of the person performing the action or sign to be recognized. The positions of the joints in the skeletons are then used to create signals. The created signals are the input for the Common Spatial Patterns algorithm.

The Common Spatial Patterns (CSP) algorithm [54], a mathematical technique for signal processing, has been widely used in Brain Computer Interface (BCI) applications for electroencephalography (EEG) systems [55], [56]. It has also been applied in the field of electrocardiography (ECG) [57], electromyography (EMG) [58], [59] or even in astronomical images for planet detection [60]. CSP was presented as an extension of Principal Component Analysis (PCA) and it consists of finding an optimum spatial filter which reduces the dimensionality of the original signals. Considering just two different classes, a CSP filter (1) maximizes the difference of the variances between the classes, maximizing the variance of filtered signals of EEG of one of the targets while minimizing the variance for the other.

$$\mathbf{W} = \operatorname{argmax} \frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad (1)$$

As the feature vectors of the spatial filter  $W$  are sorted by variance, the first and the last  $q$  vectors, which produce the smallest variance for one class and the largest variance for the other class, are used to project the original signals (2). Finally, the feature vector is obtained by calculating the variance of the transformed signals  $Z$  (3). The feature vector value for the  $p$ -th component of the  $i$ -th trial is the logarithm of the normalized variance.

$$Z = W^T X \quad (2)$$

$$f_p^i = \log \left( \frac{\operatorname{var}_p(Z_i)}{\sum_{p=1}^{2q} \operatorname{var}_p(Z_i)} \right) \quad (3)$$

The CSP algorithm can only work with pairs of classes, but multiclass classification is possible using pairwise classification approaches, such as One versus One (OVO) as a class binarization technique [61].

The CSP-filtered signals are further processed applying two matrix operations. Being  $M \in R^{K \times L}$  a matrix formed by the extracted video signals where  $K$  is the number of signals and  $L$  is the maximum length value, on the one hand, a matrix

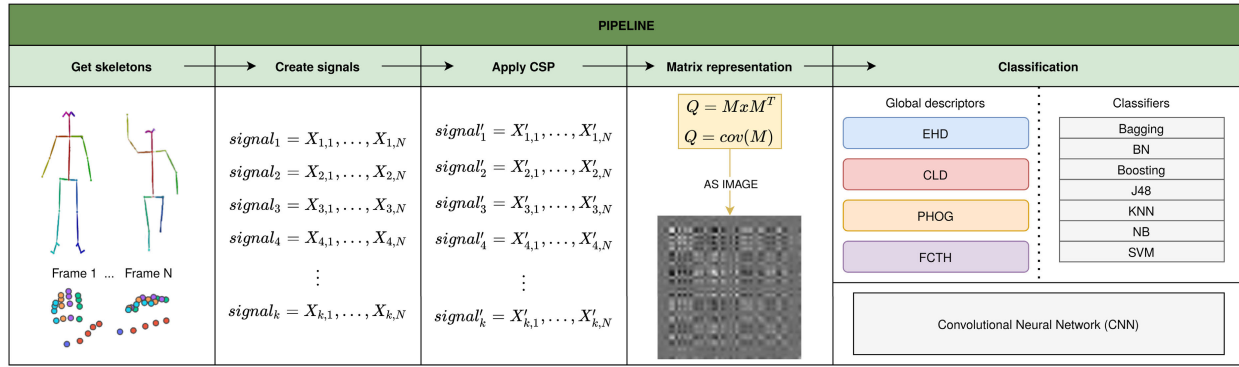


FIGURE 2. Overview of the presented approach.

multiplication is performed (Eq. 4) and, on the other hand, the covariance matrix is calculated (Eq. 5). The motivation behind these transformations is that one of the dimensions of the matrix representing the signals is the number of signals, but the other could be arbitrary long, as it is the number of time steps or frames. Therefore a matrix multiplication by its transpose reduces the data to a manageable size. On the other side, centering a matrix, multiplying by its transpose and dividing by the number of rows - 1 produces the covariance matrix, which provides information about global characteristics of the signals.

$$Q = M * M^T \quad (4)$$

$$Q = cov(M) = \frac{1}{n-1} \sum_{j=1}^n (M_j - \bar{M})(M_j - \bar{M})^T \quad (5)$$

A  $K \times K$  matrix is obtained, being  $K$  the number of signals, the number of rows of the matrix. These matrices are then treated as images; that is, for each video one image is obtained.

The created images are then classified to identify the action that has been performed on the original video.

## IV. EXPERIMENTAL SETUP

### A. DATA SETS

In the experiments presented in this paper two data sets have been used: one has been collected by us, and the other is a public available database.

#### 1) ACTION RECOGNITION (AR)

This database has been created by recording videos with the camera of the semi-humanoid robot Pepper. It consists of 272 videos with 6 action categories. There are around 45 clips in each category, performed by 46 different people. When recording the actions, the robot adjusts the orientation of its head according to the location of the face of the person appearing in its field of view.

The action categories and the information about the videos can be seen in Table 1.

These are the 6 categories that are included in the data set:

- 1) Come: gesture for telling the robot to come to you.
- 2) Five: gesture of 'high five'.

TABLE 1. Characteristics of each action category.

Category	#video	Resolution	FPS
Come	46	320×480	10
Five	45	320×480	10
Handshake	45	320×480	10
Hello	44	320×480	10
Ignore	46	320×480	10
Look at	46	320×480	10

TABLE 2. LSA64 signs used for classification and their characteristics.

CLASS	ID	ENV.	CLASS	ID	ENV.	CLASS	ID	ENV.
Opaque	001	Indoor	Born	015	Indoor	Birthday	030	Outdoor
Red	002	Indoor	Learn	016	Indoor	Hungry	033	Outdoor
Green	003	Indoor	Call	017	Indoor	Ship	037	Outdoor
Yellow	004	Indoor	Skimmer	018	Indoor	None	038	Outdoor
Bright	005	Indoor	Bitter	019	Indoor	Name	039	Outdoor
Light-blue	006	Indoor	Sweet milk	020	Indoor	Patience	040	Outdoor
Colors	007	Indoor	Milk	021	Indoor	Perfume	041	Outdoor
Red2	008	Indoor	Water	022	Indoor	Deaf	042	Outdoor
Women	009	Indoor	Food	023	Indoor	Candy	046	Outdoor
Enemy	010	Indoor	Argentina	024	Outdoor	Cheewing-gum	047	Outdoor
Son	011	Indoor	Uruguay	025	Outdoor	Shut down	052	Outdoor
Man	012	Indoor	Country	026	Outdoor	Buy	059	Outdoor
Away	013	Indoor	Last name	027	Outdoor	Realize	062	Outdoor
Drawer	014	Indoor	Where	028	Outdoor	Find	064	Outdoor
FPS: 60			Resolution: 1920x1080			Pos. camera: 2m away		

- 3) Handshake: gesture of handshaking with the robot.
- 4) Hello: gesture for telling hello to the robot.
- 5) Ignore: ignore the robot, pass by.
- 6) Look at: stare at the robot in front of it.

#### 2) SIGN LANGUAGE RECOGNITION(SLR)

For the SLR task an Argentinian Sign Language (LSA) data set, LSA64 data set [62] is used, which is composed of 64 different LSA signs. The videos were recorded by 10 non-expert subjects, who repeat each sign 5 times. Among the performed signs, both one-handed (42 signs performed with the right hand) and two-handed (22 signs) signs can be found. In order to simplify the classification problem, a subset of the data set has been selected, precisely the 42 one-handed videos have been used. The name and information of the used signs can be seen in Table 2. Thus, the subset used is composed by 2100 videos, where 1150 videos were recorded outdoors with natural lighting (23 signs, 10 signers, 5 repetitions) and 950 videos were recorded indoors with artificial lighting (19 signs, 10 signers, 5 repetitions).

The signers wore black clothes and colored gloves (red and green), and they were recorded with a white wall as background. The colored gloves (red and green) are used in order to facilitate the task of hand segmentation, although this

is not helpful in the approach presented in this paper, as no hand segmentation is performed. It must be mentioned that the subjects do not make use of the facial expression when performing the signs, they just focus on the movements of the hands.

**B. METHOD APPLICATION**

The method described in Section III has been applied to both of the presented data sets. Even though both data sets correspond to scenarios where the action or the sign performed by the person in front of the camera needs to be identified, some differences have been made on the application of the method. Different classifiers have also been tested on the classification step of the images that are created from the videos.

The different setups that have been tried are described below.

**1) GET SKELETONS AND CREATE SIGNALS**

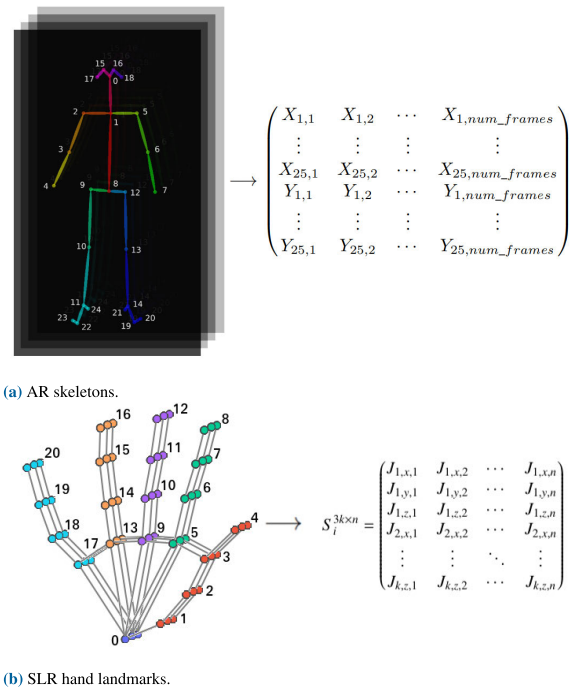
The selected data sets have different purposes; on one hand the AR data set is an action recognition data set where different subjects perform general actions where the whole body is involved. On the other hand, on the SLR data set the focus is always on the upper body of the signers, specially on their hands. Due to this dissimilarity, different methods have been selected to extract the skeletal information of the videos of the different databases.

On the AR data set, it has been decided to use OpenPose [5] to extract the skeletons of the people of the scene. This tool is a real-time multi-person system to detect human body on single images. In this case, the actions that have to be recognized are centered in the actor who perform them. Therefore, the skeleton of the actor has been extracted in every frame of each video. The system has been designed with the restriction that only a person ought to be in the field of view of the camera. In any case, as OpenPose allows for restricting the detection to only one person in order to speed up the processing and tracking, this approach ignores people in the background.

OpenPose returns the  $(x, y)$  positions of 25-keypoints (joints). After obtaining the skeleton information for every frame of each video, we can create 50 different signals to represent each video, where each signal will be the position of a skeleton keypoint over time. This way, there will be 50 signals (25 for the  $x$  position of the joints and another 25 for the  $y$  positions) with the same length as the original video (one skeleton per frame). The appearance of the skeleton and the matrix extracted from the skeletons can be seen in Fig. 3a.

For the SLR data set a technology called MediaPipe [6] has been used to track the positions of the hands in each frame of the video. More precisely the MediaPipe Holistic solution is used, which integrates separate models for pose, face and hand components. This solution offers a real-time hand tracking, which includes 21 hand landmarks for each hand.

It has been noticed that due to the speed of the movements or the use of color gloves, MediaPipe is not able to track the



**FIGURE 3. Joints positions and matrix representation of the extracted signals.**

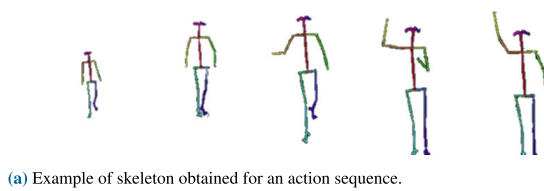
hands in 52 videos of the SLR data set. In order to try to solve this issue, the original videos have been converted from RGB color space to black and white. This way, the performance of Mediapipe has been improved and the number of videos where the hand is not detected in any frame has dropped to 6.

Each landmark returned by MediaPipe is composed of three coordinates  $(x, y, z)$ , where  $(x, y)$  denote its position and the  $z$  coordinate represents the depth of each joint in reference to the position of the wrist. Once the landmark values are obtained, a set of signals is created for every video of the database.

In Fig. 3b a graphical explanation of the hand landmarks and the extracted set of signals  $S$  for video  $i$  are shown, where  $k$  is the number of joint features,  $n$  is the number of frames and  $J_{u,c,v}$  is the landmark value for joint  $u$ , coordinate  $c : x, y, z$  and frame  $v$ . For each frame 21 joints ( $k = 21$ ) are extracted, and as each landmark is composed of  $(x, y, z)$  values, the signal matrix has 63 rows: 3 values  $(x, y, z)$  for each one of the 21 joints ( $3 \times 21 = 63$ ). In Fig. 4 an example of the sequence obtained from a video is shown, both for the action recognition data set and the LSA64 data set. In 4a the skeleton obtained by OpenPose is presented and, in 4b, the hand landmarks extracted with MediaPipe.

**2) APPLY THE COMMON SPATIAL PATTERNS ALGORITHM**

In order to compute the CSP algorithm, the signals have been preprocessed first. On the one hand, it has to be considered that some joints could be missing from the captured skeletons when the actor does not fit entirely in the camera range or OpenPose and MediaPipe are not able to capture some of the landmarks. In these cases, the missing joints values are estimated by a linear interpolation, using the previous

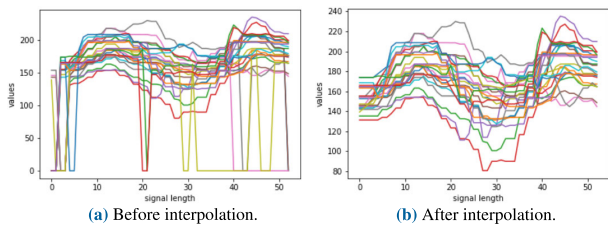


(a) Example of skeleton obtained for an action sequence.



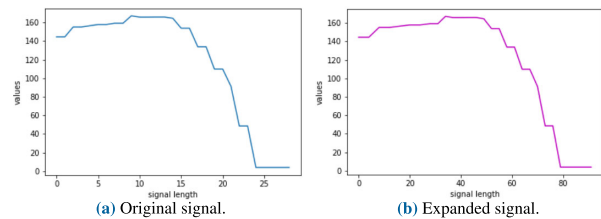
(b) Example of hand landmarks obtained for a sign sequence.

FIGURE 4. Frame sequences examples for different categories.



(a) Before interpolation. (b) After interpolation.

FIGURE 5. Interpolation example to avoid missing values.



(a) Original signal. (b) Expanded signal.

FIGURE 6. Interpolation example to enlarge signals.

and next values of that joint. The interpolation is done to avoid having missing values, and assuming that consecutive values of joints positions follow a smooth curve. An example of 25 signals of the  $x$  poses of a joint can be seen in Fig. 5, where the signals before and after the interpolation are shown.

Furthermore, all the signals of every video have been set to the same length. Since this is not the case of the videos used in the experiments, the longest video has been selected and all the signals have been enlarged to the number of frames of that video. To assign the same length to all the videos, new values have been introduced between the original values of the joints, uniformly. The added values are interpolated with the original values among which they are found. An example of a single signal extension is shown in Fig. 6.

Once the landmarks are processed and, hence, the signals are formed, the CSP is computed in order to separate the classes according to their variance. Since in both data sets a multiclass classification needs to be performed, a pairwise approach is used. In Fig. 7 an example of the variances obtained from the signals transformed applying the CSP algorithm can be seen.

As it has been explained, the CSP filter tries to separate the given classes by variance, where the first  $q$  vectors produce

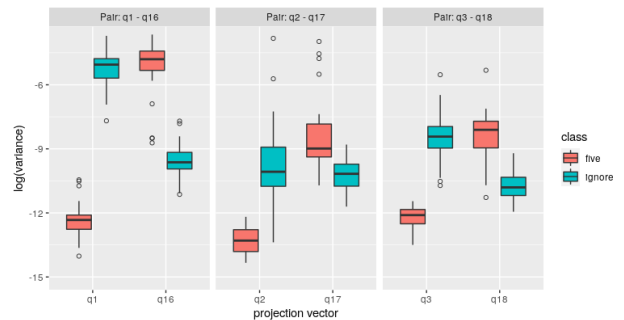
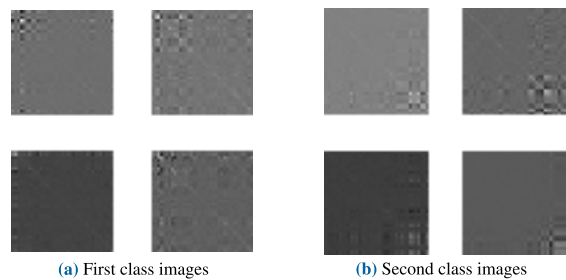


FIGURE 7. Boxplot of variances obtained from different projection vectors, by class.



(a) First class images (b) Second class images

FIGURE 8. Examples of achieved images, after applying CSP and matrix operation. On the left images for class come are shown and on the right, images for class five.

the smallest variance for one class and the largest for the other, while the last  $q$  vectors produce the opposite. In Fig. 7, three pairs of vectors are shown ( $q1 - q16$ ,  $q2 - q17$  and  $q3 - q18$ ) and it is clearly noticeable the difference between the variances of the classes (*ignore* and *five*) in each of the vectors, where the first  $q$  vectors ( $q = 15$  in this case) minimize the variance of class *five* and maximize the variance of class *ignore*, and the last  $q$  minimize the variance of class *ignore* and maximize the variance of class *five*.

### 3) MATRIX REPRESENTATION

In Fig. 8 some examples of obtained images are shown, for the classes *come* and *five* of the presented AR data set (as mentioned before, all the process is computed in pairs). The images are low-dimensional since they come from  $50 \times 50$  matrices.

### 4) CLASSIFICATION

Once the summary images are created, different classification strategies can be used in order to classify them into the original action classes. In this work two paths have been explored; a global descriptor strategy and the use of a Convolutional Neural Network (CNN).

#### a: VISUAL GLOBAL DESCRIPTORS

Some commonly used visual descriptors (image descriptors) have been used to extract useful information from the created summary images. These descriptors describe visual features of images or videos, encoding interesting information into a list of numbers. They describe basic characteristics such as shape, color, texture or motion. In our approach four different descriptors have been used:

- The Color Layout Descriptor (CLD): a technique, proposed by the MPEG-7 standard, designed to capture the spatial distribution of color of an image.
- The Pyramid Histogram of Oriented Gradients (PHOG) descriptor [63]: it represents an image by its local shape and the spatial information of the shape.
- The Fuzzy Color and Texture Histogram (FCTH) descriptor [64]: it joins color and texture information in a single histogram.
- The Edge Histogram Descriptor (EHD) [65]: it extracts MPEG-7 edge histogram features from images, a summary of the edges directions across an image.

Different classifiers have been trained with the feature vectors constructed from the descriptors. These classifiers are: Bagging, Bayesian Network (BN), Boosting, J48 classification Tree, K-Nearest Neighbors (KNN), Naive Bayes (NB) and Support Vector Machine (SVM). For each descriptor type these seven classifiers have been trained and evaluated by a 10-fold Cross Validation.

#### b: CONVOLUTIONAL NEURAL NETWORKS

Additionally, a Convolutional Neural Network (CNN) has been applied to classify the summary images obtained for each video. Its performance might drastically vary between several hyperparameter configurations, and therefore, in order to provide a fair comparison, we have used Keras Tuner Hypermodel, with a RandomSearch tuner to look for good configurations automatically. The input is composed by one image per video with a shape of  $50 \times 50 \times 1$  in the case of the AR data set and  $63 \times 63 \times 1$  when it refers to LSA64 database, since the images are gray-scale.

Convolutional layers, dropout layers, max pooling layers and a final dense layer of two units (as the classification is performed by pairs) make up the network. Adam is used as optimizer and categorical cross-entropy as loss function. The learning rate, activation functions, number of filters and dropout rate hyperparameters have been tuned.

#### 5) COMPARISON

To finish, the proposed approach is compared with a type of neural network widely used for video action recognition task, a Long Short-Term Memory (LSTM) network. The LSTM network has the signals obtained from the joints of the skeleton as input, so its input is bidimensional (number of frames, number of joints) and the output is of 64 units. Then a dense layer of 2 units has been placed, since the classification is carried out between two classes. Regarding the rest of the hyperparameters, Adam optimizer and categorical cross-entropy loss function have been used, and the network has been trained for 100 epochs with a batch size of 25.

#### V. EXPERIMENTAL RESULTS

In this section the obtained results for the experimentation that has been carried out are presented. First, the results obtained for the action recognition data set are shown

and afterwards the outcomes obtained for the LSA64 are explained.

#### A. ACTION RECOGNITION DATA SET

Table 3 and 4 show the results obtained using the characteristics extracted by the global descriptors from the images obtained by matrix multiplication (equation 4) and from the images of covariance matrices (equation 5), respectively. A mean accuracy value is also presented for each pair of classes, with the best values highlighted in bold. In the tables, to summarize, the classes have been represented as follows: C (come), F (five), H (handshake), He (hello), I (ignore) and L (look at).

First, if the two tables (Table 3 and Table 4) are compared with each other, both the matrix multiplication and the covariance matrix obtain good results. After applying the descriptors, both representations yield a mean over all the entries of  $\sim 0.86$  and a median of  $\sim 0.92$ . Thus, in general the results are encouraging.

Next, three types of comparisons are made: by classifier type, by image descriptor type, and by class pairs.

Regarding the classifiers, on average they all get even results, there is no one that stands out from the rest. Even so, it could be said that Naive Bayes (NB) has been the worst of all in both representations and the best average result is achieved by Support Vector Machine (SVM).

Concerning the descriptors, the difference is more noticeable. EHD and PHOG get outstanding results with an average accuracy of  $\sim 0.95$ . The CLD descriptor does not get bad results either ( $\sim 0.87$  on average). The worst results, by far, are achieved with the characteristics obtained using the FCTH descriptor.

Finally, in relation to the pairs of classes, the good results of *handshake-hello (H-He)*, *handshake-ignore (H-I)* or *hello-look at (He-L)* can be highlighted. For instance, the pairs *five-ignore (F-I)* and *come-ignore (C-I)* achieve very good results with all the descriptors except FCTH, which as it has been already mentioned is the descriptor with the worst results overall. However, the worst pair of classes (*five-hello (F-He)*) obtains an average of 0.71 accuracy, therefore very good results have been achieved in the experiment.

Table 5 shows the average accuracy values obtained for each type of descriptor and image of the presented approach, along with the results obtained using the CNN network taking as input the matrix representation images and the results achieved by the LSTM mentioned before, where the best values are highlighted in boldface. The results obtained with a previous approach [2] are also shown.

The obtained accuracy values show that our new approach beats LSTM method and the previous approach for every class pairs. Furthermore, observing this table it is evident that the best results are achieved using the CNN and, EHD or PHOG when it comes to global descriptors. Although the best mean value corresponds to the use of PHOG descriptor, in 9 out of 15 pair of classes CNN performs better. Regarding the type of images, generally better outcomes are obtained using





TABLE 5. Comparison between presented approaches, previous approach and a LSTM network for AR data set.

Pair of Categories	Matrix multiplication					Covariance matrix					LSTM	Previous approach
	CLD	EHD	FCTH	PHOG	CNN	CLD	EHD	FCTH	PHOG	CNN		
COME-FIVE	0.6955	0.8461	0.7849	0.9278	0.9355	0.7253	0.8618	0.7739	0.9231	<b>0.9677</b>	0.8628	0.7579
COME-HANDSHAKE	0.7896	0.8964	0.7190	0.9654	0.8710	0.8069	0.9200	0.7096	0.9246	<b>1</b>	0.7739	0.8668
COME-HELLO	0.7460	0.7587	0.6397	0.9063	<b>1</b>	0.7809	0.8048	0.7159	0.8746	0.9333	0.7336	0.5334
COME-IGNORE	0.9643	<b>0.9984</b>	0.5481	0.9891	0.9677	0.9829	0.9953	0.5466	0.9953	0.9355	0.9575	0.9779
COME-LOOK_AT	0.8463	0.9705	0.7624	0.9876	<b>1</b>	0.9192	0.9658	0.7220	0.9658	<b>1</b>	0.7849	0.8678
FIVE-HANDSHAKE	0.8413	0.9699	0.4809	0.9635	0.8667	0.8619	0.9762	0.5111	0.9540	<b>1</b>	0.8125	0.9557
FIVE-HELLO	0.7641	0.6661	0.5457	0.8732	<b>0.9333</b>	0.7062	0.7175	0.5907	0.8443	<b>0.9333</b>	0.9125	0.8208
FIVE-IGNORE	0.9215	0.9953	0.4694	0.9969	<b>1</b>	0.9733	<b>1</b>	0.4741	0.9953	<b>1</b>	0.9789	0.9668
FIVE-LOOK_AT	0.9011	0.9937	0.7127	0.9749	<b>1</b>	0.8775	0.9953	0.6389	0.9812	0.9677	0.8889	0.9667
HANDSHAKE-HELLO	0.8588	0.9647	0.9310	0.9663	0.8667	0.8668	0.9663	0.9390	<b>0.9776</b>	0.8333	0.7108	0.7431
HANDSHAKE-IGNORE	0.9560	0.9859	0.8399	<b>0.9953</b>	0.9677	0.9372	0.9843	0.8932	<b>0.9953</b>	0.9032	0.9764	0.9889
HANDSHAKE-LOOK_AT	0.8964	0.9090	0.6295	0.9466	0.9355	0.8917	0.9294	0.6075	0.9372	<b>0.9677</b>	0.8350	0.8235
HELLO-IGNORE	0.9667	0.9905	0.5063	0.9984	0.9667	0.9492	0.9857	0.5413	<b>1</b>	0.9667	0.9789	0.9333
HELLO-LOOK_AT	0.9524	<b>0.9937</b>	0.9079	0.9889	0.9333	0.9127	0.9873	0.8841	0.9587	0.9333	0.5733	0.8445
IGNORE-LOOK_AT	0.9721	0.9938	0.6413	<b>0.9984</b>	0.8387	0.9488	0.9798	0.5978	0.9922	0.8710	0.9775	0.9889
MEAN	0.8715	0.9288	0.6746	<b>0.9652</b>	0.9389	0.8760	0.9380	0.6764	0.9546	0.9475	0.8505	0.8691

TABLE 6. Results obtained with matrix multiplication and visual global descriptors approach for LSA64 data set.

Descrip.	Classif.	MIN	Q1	MEAN	MEDIAN	Q3	MAX
CLD	Bag.	0.5263	0.8300	0.8770	0.8900	0.9424	1
	BN	0.4800	0.8300	0.8801	0.9000	0.9500	1
	Boost.	0.5600	0.8400	0.8847	0.9000	0.9500	1
	J48	0.5500	0.8400	0.8812	0.8900	0.9400	1
	KNN	0.5400	0.8300	0.8773	0.8900	0.9495	1
	NB	0.4900	0.8469	0.8918	0.9000	0.9500	1
EHD	SVM	0.5800	0.8400	0.8855	0.9000	0.9500	1
	Bag.	0.5600	0.9800	0.9837	0.9900	1	1
	BN	0.5000	0.9900	<b>0.9911</b>	1	1	1
	Boost.	0.5500	0.9900	0.9883	0.9900	1	1
	J48	0.5700	0.9800	0.9846	0.9900	1	1
	KNN	0.5700	0.9800	0.9822	0.9900	1	1
FCTH	NB	0.5600	0.9900	0.9893	1	1	1
	SVM	0.5400	0.9900	0.9889	1	1	1
	Bag.	0.3434	0.6176	0.7120	0.7100	0.8100	1
	BN	0.4700	0.5000	0.6811	0.6900	0.8025	1
	Boost.	0.4000	0.6200	0.7139	0.7100	0.8100	1
	J48	0.3800	0.6200	0.7140	0.7100	0.8100	1
PHOG	KNN	0.3434	0.6100	0.7103	0.7100	0.8101	1
	NB	0.3200	0.6100	0.7024	0.7000	0.7900	1
	SVM	0.3900	0.6100	0.7070	0.7000	0.8000	1
	Bag.	0.5200	0.9400	0.9530	0.9800	0.9900	1
	BN	0.4900	0.9700	0.9753	0.9900	1	1
	Boost.	0.4500	0.9700	0.9718	0.9900	1	1
PHOG	J48	0.4900	0.9600	0.9686	0.9900	1	1
	KNN	0.5300	0.9400	0.9541	0.9800	1	1
	NB	0.5300	0.9500	0.9598	0.9800	0.9900	1
	SVM	0.5300	0.9700	0.9726	0.9900	1	1

Comparing both tables (Table 6 and Table 7) better results are obtained when using the matrix multiplication. However, there is not a noticeable difference between them.

Regarding the used descriptors, there is a clear difference between the outcomes obtained with each one of them. EHD descriptor is the one which achieves better results. In Table 6 the mean accuracy values are over 0.98 and the 75% of the pairs of classes obtain higher than 0.98 accuracy values. In Table 7 the mean accuracy values are greater than 0.97 and the Q1 value indicates that the 75% of the pairs of classes achieves at least a 0.96 accuracy value.

The PHOG descriptor also obtains good results. Many pairs of classes obtain a 100% of accuracy and the mean values are ~0.95 for both matrix representation methods. The less suitable descriptor is FCTH. The results obtained with the features extracted with this descriptor are the lowest, where the mean values are ~0.7, which shows a great difference when comparing with the others.

TABLE 7. Results obtained with covariance matrix and visual global descriptors approach for LSA64 data set.

Descrip.	Classif.	MIN	Q1	MEAN	MEDIAN	Q3	MAX
CLD	Bag.	0.5263	0.7696	0.8309	0.8400	0.9100	1
	BN	0.4700	0.7600	0.8261	0.8586	0.9200	1
	Boost.	0.5100	0.7800	0.8419	0.8500	0.9194	1
	J48	0.5100	0.7800	0.8404	0.8500	0.9100	1
	KNN	0.5053	0.7600	0.8292	0.8400	0.9100	1
	NB	0.4600	0.7900	0.8517	0.8700	0.9200	1
EHD	SVM	0.4400	0.7800	0.8413	0.8500	0.9100	1
	Bag.	0.7100	0.9697	0.9750	0.9800	0.9900	1
	BN	0.7500	0.9800	0.9868	0.9900	1	1
	Boost.	0.7600	0.9800	0.9859	0.9900	1	1
	J48	0.7400	0.9684	0.9778	0.9895	1	1
	KNN	0.7400	0.9600	0.9722	0.9800	0.9900	1
FCTH	NB	0.7200	0.9800	0.9860	0.9900	1	1
	SVM	0.6600	0.9895	<b>0.9878</b>	1	1	1
	Bag.	0.4200	0.6000	0.7017	0.6900	0.8000	1
	BN	0.4600	0.5000	0.6645	0.6667	0.7980	1
	Boost.	0.4100	0.6100	0.7042	0.7000	0.8081	1
	J48	0.4343	0.6100	0.7027	0.6900	0.8000	1
PHOG	KNN	0.3600	0.6000	0.6998	0.6900	0.7985	1
	NB	0.3700	0.6000	0.6899	0.6737	0.7700	1
	SVM	0.3700	0.6000	0.6933	0.6737	0.7900	1
	Bag.	0.5400	0.8800	0.9159	0.9500	0.9800	1
	BN	0.4949	0.9300	0.9443	0.9800	0.9900	1
	Boost.	0.4900	0.9200	0.9400	0.9700	0.9900	1
PHOG	J48	0.5053	0.9100	0.9344	0.9700	0.9900	1
	KNN	0.5100	0.8900	0.9170	0.9500	0.9800	1
	NB	0.5100	0.8900	0.9219	0.9600	0.9800	1
	SVM	0.4848	0.9200	0.9421	0.9700	0.9900	1

Concerning the classifiers, there is not a perceptible contrast between them. As mentioned before, the best mean values have been obtained with BN and SVM. However, the worst average values have also been obtained with the BN classifier and FCTH descriptor. It can be concluded that their performance depends on the configuration used before the classification.

In order to compare the differences between the tested classes, in Table 8 the mean values obtained for each class of the data set are shown. These values have been calculated with the accuracy values of all the test pairs in which each class has participated. These mean values are achieved with the best configuration, in this case, the features obtained after applying the EHD descriptor to the images obtained by the matrix multiplication and performing the classification with a Bayesian Network.

All the classes obtain a mean accuracy value between 0.97 and 1.00. Therefore, not many conclusions can be drawn about the difference of classes, since all of them obtain good

**TABLE 8.** Mean accuracy values obtained with the best configuration (Matrix Multiplication, EHD descriptor and BN classifier) for each class of LSA64 data set.

Opaque	Red	Green	Yellow	Bright	Light-blue
0.9968	0.9868	0.9958	0.9941	0.9973	0.9932
Colors	Red 2	Women	Enemy	Son	Man
0.9824	0.9753	0.9875	0.9954	0.9932	0.9921
Away	Drawer	Born	Learn	Call	Skimmer
0.9946	0.9893	0.9971	0.9951	0.9934	0.9963
Bitter	Sweet-milk	Milk	Water	Food	Argentina
0.9815	0.9932	0.9844	0.9900	0.9876	0.9961
Uruguay	Country	Last name	Where	Birthday	Hungry
0.9929	0.9934	0.9849	0.9759	0.9910	0.9941
Ship	None	Name	Patience	Perfume	Deaf
0.9893	0.9953	0.9868	0.9959	0.9909	0.9912
Candy	Chewing-gum	Shut down	Buy	Realize	Find
0.9973	0.9873	0.9971	0.9878	0.9938	0.9912

**TABLE 9.** Comparison between presented approaches and LSTM for SLR.

	Matrix multiplication			Covariance matrix			LSTM
	EHD	PHOG	CNN	EHD	PHOG	CNN	
MIN	0.5000	0.4500	0.5455	0.6600	0.4848	0.5000	0.6100
Q1	0.9899	0.9600	0.8788	0.9700	0.9100	0.8788	0.8900
MEAN	<b>0.9869</b>	0.9650	0.9267	0.9816	0.9308	0.9171	0.9186
MEDIAN	1	0.9900	0.9394	0.9900	0.9600	0.9394	0.9300
Q3	1	1	1	1	0.9900	0.9697	0.9600
MAX	1	1	1	1	1	1	1

results. *Bright*, *Born*, *Candy* and *Shut down* classes get the highest values and classes like *Red2*, *Where* and *Bitter* get slightly worse values.

A comparison with a LSTM network is performed and the results are shown in Table 9. Some statistics of the obtained accuracy values are displayed, which refer to all pairs of classes. For the comparison, it has been decided to show only the results obtained with EHD and PHOG descriptors, as they are the ones which performed best. The results achieved by applying a CNN after creating the images are also presented.

Although the minimum obtained value among all the pairs of classes is lower in the presented approaches, the rest of the statistics show that better accuracy values are obtained than with the LSTM method. While the LSTM method obtains an average of 0.9186, our approach achieves a mean value of 0.9869. Regarding the approaches which use global descriptors to extract characteristics from the images to train the classifiers, both the median and the Q1 and Q3 values indicate that a greater number of pairs of classes obtain better results than with the LSTM.

All of the 4 configurations presented in the table are able to surpass the LSTM method, being the EHD descriptor the one that suits best, as mentioned above. When using the CNN, although the median and Q3 values are higher, there are several pairs of classes that achieve lower results than with the LSTM, as indicated by the minimum and Q1 values.

Finally, and in order to understand these results better, Table 10 shows the mean of the values obtained for each class using the Convolutional Neural Network with both types of images (MM: Matrix Multiplication, COV: covariance matrix).

All the classes obtain high mean accuracy values, which vary between 0.85 and - 0.96. For the matrix multiplication images the best value is obtained by the *Sweet-milk* class (0.9601) and the worst by *Find* (0.8853), whereas for the covariance matrix images the best value is obtained by the

**TABLE 10.** Mean accuracy values obtained for each class of LSA64 data set with CNN.

	Opaque	Red	Green	Yellow	Bright	Light-blue
MM	0.9069	0.9475	0.9209	0.9038	0.9401	0.9438
COV	0.9504	0.9283	0.9208	0.9075	0.9416	0.9209
	Colors	Red 2	Women	Enemy	Son	Man
MM	0.9157	0.9259	0.9356	0.9186	0.9149	0.9319
COV	0.9216	0.8703	0.9416	0.9193	0.9430	0.8868
	Away	Drawer	Born	Learn	Call	Skimmer
MM	0.9119	0.9261	0.9208	0.9231	0.9061	0.9379
COV	0.9290	0.9349	0.8831	0.9141	0.9157	0.9372
	Bitter	Sweet-milk	Milk	Water	Food	Argentina
MM	0.9091	0.9601	0.9172	0.9222	0.9348	0.9341
COV	0.9148	0.9246	0.9111	0.9331	0.8854	0.9172
	Uruguay	Country	Last name	Where	Birthday	Hungry
MM	0.9453	0.9387	0.9401	0.9483	0.9172	0.9194
COV	0.9335	0.9275	0.9157	0.9067	0.9163	0.9105
	Ship	None	Name	Patience	Perfume	Deaf
MM	0.9460	0.9326	0.9164	0.9216	0.9261	0.9200
COV	0.9231	0.9260	0.9238	0.9319	0.9305	0.9194
	Candy	Chewing-gum	Shut down	Buy	Realize	Find
MM	0.9334	0.9326	0.9318	0.9446	0.9131	0.8853
COV	0.9297	0.9083	0.9097	0.8934	0.8788	0.9128

*Opaque* class (0.9504) and the worst by the *Red2* class (0.8703). In short, the results obtained for all of the classes are similar and no matrix method stands out.

## VI. CONCLUSION AND FUTURE WORK

In this paper a new pipeline for action recognition is presented, which has been applied to two different tasks in this domain: activity recognition and sign language recognition. In the presented approach the Common Spatial Patterns method has been applied to signals created from the positions of the skeleton joints of people performing different actions or signs. From the output of the method some images have been created, which have been then classified. In the classification step two approaches have been tested; one based on Visual Global Descriptors and the other a CNN implementation. The obtained results have been compared to a previous approach by the same authors and also to those obtained by a LSTM, a well-known deep learning method.

As further work, we plan to extend the range of human activities, as well as to implement the presented method in the actual robot. This would allow the robot to react to different actions performed in front of it, or to communicate with people with hearing impairments. Applications in Social Robotics are also to be developed, being this the next envisaged step.

Concerning the sign language recognition, several steps have been identified that would improve the presented method. Facial information of the signers should be added, since it is a crucial feature when interpreting sign language. Signs which use both hands should also be considered, in order to make the recognition system more complete.

On the classification step, other image descriptors could also be used, and in that case a feature subset selection step could be advisable.

## ACKNOWLEDGMENT

The authors would like to thank the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## REFERENCES

- [1] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, and B. Sierra, "Sign language recognition by means of common spatial patterns," in *Proc. 5th Int. Conf. Mach. Learn. Soft Comput. (ICMLSC)*, Jan. 2021, pp. 96–102.
- [2] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, I. Rodríguez-Rodríguez, and B. Sierra, "Shedding light on people action recognition in social robotics by means of common spatial patterns," *Sensors*, vol. 20, no. 8, p. 2436, Apr. 2020.
- [3] K. Keng Ang, Z. Yang Chin, H. Zhang, and C. Guan, "Filter bank common spatial pattern (FBCSP) in brain-computer interface," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intelligence)*, Jun. 2008, pp. 2390–2397.
- [4] S. Sethi, R. Upadhyay, and H. S. Singh, "Stockwell-common spatial pattern technique for motor imagery-based brain computer interface design," *Comput. Electr. Eng.*, vol. 71, pp. 492–504, Oct. 2018.
- [5] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.
- [6] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A framework for building perception pipelines," 2019, *arXiv:1906.08172*. [Online]. Available: <http://arxiv.org/abs/1906.08172>
- [7] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.
- [8] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, 2004, pp. 32–36.
- [9] J. C. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.
- [10] C.-C. Chen and J. K. Aggarwal, "Recognizing human action from a far field of view," in *Proc. Workshop Motion Video Comput. (WMVC)*, Dec. 2009, pp. 1–7.
- [11] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and Binet–Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1932–1939.
- [12] F. Ercis, "Comparison of histogram of oriented optical flow based action recognition methods," M.S. thesis, Dept. Elect. Electron. Eng., Middle East Tech. Univ., Ankara, Turkey, 2012.
- [13] K. Lertniphonphan, S. Aramvith, and T. H. Chalidabhongse, "Human action recognition using direction histograms of optical flow," in *Proc. 11th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Oct. 2011, pp. 574–579.
- [14] S. Akpınar and F. N. Alpaslan, "Video action recognition using an optical flow based representation," in *Proc. Int. Conf. Image Process., Comput. Vis., Pattern Recognit. (IPCV)*. Las Vegas, NV, USA: The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2014, p. 1.
- [15] S. Satyamurthi, J. Tian, and M. C. H. Chua, "Action recognition using multi-directional projected depth motion maps," *J. Ambient Intell. Hum. Comput.*, pp. 1–7, Nov. 2018.
- [16] M. Liu, H. Liu, and C. Chen, "Robust 3D action recognition through sampling local appearances and global distributions," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 1932–1947, Aug. 2018.
- [17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
- [18] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
- [19] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.
- [20] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream ConvNets," 2015, *arXiv:1507.02159*. [Online]. Available: <http://arxiv.org/abs/1507.02159>
- [21] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.
- [22] S. Arif, J. Wang, T. Ul Hassan, and Z. Fei, "3D-CNN-based fused feature maps with LSTM applied to action recognition," *Future Internet*, vol. 11, no. 2, p. 42, Feb. 2019.
- [23] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 20–36.
- [24] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 803–818.
- [25] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 7083–7093.
- [26] S. Sudhakaran, S. Escalera, and O. Lanz, "Gate-shift networks for video action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1102–1111.
- [27] D. Tran, H. Wang, M. Feiszli, and L. Torresani, "Video classification with channel-separated convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5552–5561.
- [28] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou, "Temporal pyramid network for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 591–600.
- [29] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," 2016, *arXiv:1611.06067*. [Online]. Available: <http://arxiv.org/abs/1611.06067>
- [30] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "PoTion: Pose motion representation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7024–7033.
- [31] J. Ren, N. H. Reyes, A. L. C. Barczak, C. Scogings, and M. Liu, "An investigation of skeleton-based optical flow-guided features for 3D action recognition using a multi-stream CNN model," in *Proc. IEEE 3rd Int. Conf. Image. Vis. Comput. (ICIVC)*, Jun. 2018, pp. 199–203.
- [32] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 12026–12035.
- [33] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 1, pp. 131–153, 2019.
- [34] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. B. Lakulu, "A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017," *Sensors*, vol. 18, no. 7, p. 2208, 2018.
- [35] A. Wadhawan and P. Kumar, "Sign language recognition systems: A decade systematic literature review," *Arch. Comput. Methods Eng.*, vol. 28, pp. 785–813, May 2021.
- [36] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113794.
- [37] K. Kudrinko, E. Flavin, X. Zhu, and Q. Li, "Wearable sensor-based sign language recognition: A comprehensive review," *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 82–97, 2021.
- [38] P. D. Rosero-Montalvo, P. Godoy-Trujillo, E. Flores-Bosmediano, J. Carrascal-García, S. Otero-Potosi, H. Benitez-Pereira, and D. H. Peluffo-Ordóñez, "Sign language recognition based on intelligent glove using machine learning techniques," in *Proc. IEEE 3rd Ecuador Tech. Chapters Meeting (ETCM)*, Oct. 2018, pp. 1–5.
- [39] M. I. Sadek, M. N. Mikhail, and H. A. Mansour, "A new approach for designing a smart glove for Arabic sign language recognition system based on the statistical analysis of the sign language," in *Proc. 34th Nat. Radio Sci. Conf. (NRSC)*, Mar. 2017, pp. 380–388.
- [40] C. K. M. Lee, K. K. H. Ng, C.-H. Chen, H. C. W. Lau, S. Y. Chung, and T. Tsoi, "American sign language recognition and training method with recurrent neural network," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114403.
- [41] J. J. Bird, A. Ekárt, and D. R. Faria, "British sign language recognition via late fusion of computer vision and leap motion with transfer learning to American sign language," *Sensors*, vol. 20, no. 18, p. 5151, Sep. 2020.
- [42] T.-W. Chong and B.-G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors*, vol. 18, no. 10, p. 3554, Oct. 2018.
- [43] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A modified LSTM model for continuous sign language recognition using leap motion," *IEEE Sensors J.*, vol. 19, no. 16, pp. 7056–7063, Aug. 2019.

- [44] D. Li, C. R. Opazo, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Mar. 2020, pp. 1459–1469.
- [45] M. Parelli, K. Papadimitriou, G. Potamianos, G. Pavlakos, and P. Maragos, "Exploiting 3D hand pose estimation in deep learning-based sign language recognition from RGB videos," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 249–263.
- [46] M. Jebali, A. Dakhli, and M. Jemni, "Vision-based continuous sign language recognition using multimodal sensor fusion," *Evolving Syst.*, pp. 1–14, Jan. 2021.
- [47] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," in *Proc. Conf. Signal Process. Commun. Eng. Syst. (SPACES)*, Jan. 2018, pp. 194–197.
- [48] N. Basnin, L. Nahar, and M. S. Hossain, "An integrated CNN-LSTM model for Bangla lexical sign language recognition," in *Proc. Int. Conf. Trends Comput. Cogn. Eng.* Singapore: Springer, 2021, pp. 695–707.
- [49] R. Rastgoo, K. Kiani, and S. Escalera, "Hand sign language recognition using multi-view hand skeleton," *Expert Syst. Appl.*, vol. 150, Jul. 2020, Art. no. 113336.
- [50] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1311–1325, 2018.
- [51] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "Coupled HMM-based multi-sensor data fusion for sign language recognition," *Pattern Recognit. Lett.*, vol. 86, pp. 1–8, Jan. 2017.
- [52] S. G. Azar and H. Seyedarabi, "Trajectory-based recognition of dynamic Persian sign language using hidden Markov model," *Comput. Speech Lang.*, vol. 61, May 2020, Art. no. 101053.
- [53] A. Bastanfard, N. A. Rezaei, M. Mottaghizadeh, and M. Fazel, "A novel multimedia educational speech therapy system for hearing impaired children," in *Proc. Pacific-Rim Conf. Multimedia*. Berlin, Germany: Springer, 2010, pp. 705–715.
- [54] K. Fukunaga and W. L. G. Koontz, "Application of the Karhunen–Loève expansion to feature selection and ordering," *IEEE Trans. Comput.*, vol. C-19, no. 4, pp. 311–318, Apr. 1970.
- [55] Y. Wang, S. Gao, and X. Gao, "Common spatial pattern method for channel selection in motor imagery based brain-computer interface," in *Proc. IEEE Eng. Med. Biol. 27th Annu. Conf.*, Jan. 2006, pp. 5392–5395.
- [56] Q. Novi, C. Guan, T. H. Dat, and P. Xue, "Sub-band common spatial pattern (SBCSP) for brain-computer interface," in *Proc. 3rd Int. IEEE/EMBS Conf. Neural Eng.*, May 2007, pp. 204–207.
- [57] T. N. Alotaiby, S. A. Alshebeili, L. M. Aljafar, and W. M. Alsabhan, "ECG-based subject identification using common spatial pattern and SVM," *J. Sensors*, vol. 2019, pp. 1–9, Mar. 2019.
- [58] P. Kim, K.-S. Kim, and S. Kim, "Using common spatial pattern algorithm for unsupervised real-time estimation of fingertip forces from sEMG signals," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 5039–5045.
- [59] X. Li, P. Fang, L. Tian, and G. Li, "Increasing the robustness against force variation in EMG motion classification by common spatial patterns," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 406–409.
- [60] J. Shapiro, D. Savransky, J.-B. Ruffio, N. Ranganathan, and B. Macintosh, "Detecting planets from direct-imaging observations using common spatial pattern filtering," *Astron. J.*, vol. 158, no. 3, p. 125, Aug. 2019.
- [61] I. Mendiádua, J. M. Martínez-Otzeta, I. Rodríguez-Rodríguez, T. Ruiz-Vázquez, and B. Sierra, "Dynamic selection of the best base classifier in one versus one," *Knowl.-Based Syst.*, vol. 85, pp. 298–306, Sep. 2015.
- [62] F. Ronchetti, F. Quiroga, C. A. Estrebow, L. C. Lanzarini, and A. Rosete, "LSA64: An Argentinian sign language dataset," in *Proc. XXII Congreso Argentino de Ciencias de la Computación (CACIC)*, 2016, pp. 794–803.
- [63] A. Bosch, A. Zisserman, and X. Muñoz, "Representing shape with a spatial pyramid kernel," in *Proc. 6th ACM Int. Conf. Image Video Retr.*, 2007, pp. 401–408.
- [64] S. A. Chatzichristofis and Y. S. Boutalis, "FCTH: Fuzzy color and texture histogram—A low level feature for accurate image retrieval," in *Proc. 9th Int. Workshop Image Anal. Multimedia Interact. Services*, 2008, pp. 191–196.
- [65] C. S. Won, D. K. Park, and S.-J. Park, "Efficient use of MPEG-7 edge histogram descriptor," *ETRI J.*, vol. 24, no. 1, pp. 23–30, 2002.



**ITSASO RODRÍGUEZ-MORENO** received the B.Sc. and M.Sc. degrees in computer science from the University of the Basque Country, in 2018 and 2019, respectively, where she is currently pursuing the Ph.D. degree with the Department of Computer Sciences and Artificial Intelligence, under a FPU Grant from the Spanish Government.

She is currently a member of the Robotics and Autonomous Systems Group. Her research interests include machine learning, computer vision, and robotics.



**JOSÉ MARÍA MARTÍNEZ-OTZETA** received the B.Sc. and Ph.D. degrees in computer science from the University of the Basque Country, in 1993 and 2008, respectively.

He is currently a Postdoctoral Researcher with the Department of Computer Sciences and Artificial Intelligence, University of the Basque Country. He is also a member of the Robotics and Autonomous Systems Group. His research interests include machine learning, computer vision, and robotics.



**IZARO GOIENETXEA** received the B.Sc. degree in computer science from the University of the Basque Country, Spain, in 2008, and the Ph.D. degree from the Department of Computer Science and Artificial Intelligence, University of the Basque Country, in 2019.

She is currently a member of the Robotics and Autonomous Systems Research Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country.

Her current research interests include music generation and classification, computer vision, and machine learning.



**IGOR RODRIGUEZ** received the B.Sc. and Ph.D. degrees in computer science from the University of the Basque Country, in 2012 and 2018, respectively.

He is currently a member of the Robotics and Autonomous Systems Research Group, Department of Computer Science and Artificial Intelligence. His current research interests include robotics, human–robot interaction, and machine learning.



**BASILIO SIERRA** received the B.Sc. degree in computer sciences, the M.Sc. degree in computer science and architecture, and the Ph.D. degree in computer sciences from the University of the Basque Country, Donostia-San Sebastian, Spain, in 1990, 1992, and 2000, respectively.

He is currently a Full Professor with the Department of Computer Sciences and Artificial Intelligence, University of the Basque Country. He is also the Co-Director of the Robotics and Autonomous Systems Group. His research interests include robotics and machine learning, where he is working on the use of different paradigms to improve behaviors.

...

# Sign Language Recognition by Means of Common Spatial Patterns

**Title:** Sign Language Recognition by Means of Common Spatial Patterns

**Authors:** I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, B. Sierra

**Conference:** The 5th International Conference on Machine Learning and Soft Computing

**Publisher:** ACM

**DOI:** 10.1145/3453800.3453818

**Year:** 2021

**Times cited:** 3 (Google Scholar) / 3 (Scopus)



# Sign Language Recognition by Means of Common Spatial Patterns

Itsaso Rodríguez-Moreno\*  
itsaso.rodriguez@ehu.es  
University of the Basque Country (UPV/EHU)  
Donostia-San Sebastián, Spain

Izaro Goienetxea  
University of the Basque Country (UPV/EHU)  
Donostia-San Sebastián, Spain  
izaro.goienetxea@ehu.es

José María Martínez-Otzeta  
University of the Basque Country (UPV/EHU)  
Donostia-San Sebastián, Spain  
josemaria.martinez@ehu.es

Basilio Sierra  
University of the Basque Country (UPV/EHU)  
Donostia-San Sebastián, Spain  
b.sierra@ehu.es

## ABSTRACT

Currently, and despite the efforts that have been made, people with hearing impairments often have difficulties to use applications that have been designed for people who can hear, or simply to communicate with their environment. In this work, we present an Argentinian Sign Language (LSA) recognition system which distinguishes between different signs using hand landmarks extracted from the videos of the dataset. The Common Spatial Patterns (CSP) algorithm is used to extract features, and the classification is performed with multiple classifiers. Different experiments have been made from which promising results have been obtained.

## CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Activity recognition and understanding**.

## KEYWORDS

Sign Language Recognition, Computer Vision, Common Spatial Patterns

### ACM Reference Format:

Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, and Basilio Sierra. 2021. Sign Language Recognition by Means of Common Spatial Patterns. In *2021 The 5th International Conference on Machine Learning and Soft Computing (ICMLSC'21), January 29–31, 2021, Virtual Event, Vietnam*. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3453800.3453818>

## 1 INTRODUCTION

According to the World Health Organization (WHO), there are about 466 million people with hearing impairments, representing more than 5% of world population, and this number is expected to increase in the coming years. In order to be able to communicate, many people with hearing problems, or people who have trouble

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*ICMLSC'21, January 29–31, 2021, Virtual Event, Vietnam*

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8761-3/21/01...\$15.00

<https://doi.org/10.1145/3453800.3453818>

communicating through oral language, make use of sign languages. Although it is a rather commonly held belief among the general public that sign language is the same across all countries, there are more than 300 sign languages around the world.

In general, the working knowledge of sign language by non-deaf people is not widespread, which causes communication problems for users in environments without an interpreter, making the daily life interaction difficult. Therefore, to tackle these issue, different technological solutions have been developed, which involve gesture recognition and sign language linguistics, although the latter has been studied less than the former.

Sign languages are quite complex, as they are natural languages with grammatical structures, which makes them difficult to translate into spoken languages. Apart from the signs that are performed using the hands, to compose the whole message other factors such as hand location, body position or facial expression are also important. The most important source of information comes from the hands, distinguishing them between the dominant (the one that performs more complex movements and moves the most) and the non-dominant. Other important features, as mentioned before, are extracted from the facial expression, which can be more or less useful depending on the signs that are being performed. Therefore, a good Sign Language Recognition system should take all these features into account in order to perform the task correctly, but data cleaning and dimensionality reduction also play a key role.

In this paper, a new approach for video based Sign Language Recognition is presented, where a technology named MediaPipe [5] is used to extract the position of a set of joints of the moving hands that are performing signs in the videos. These positions are then used to compose a set of signals that are transformed using the Common Spatial Patterns (CSP) [3] algorithm. Even though this algorithm is widely used with EEG signals, recently it has been used in video action recognition tasks [9]. Some summarizing descriptors are obtained from the transformed signals, which are then used to perform the classification.

The rest of the paper is organized as follows. First, in Section 2 some related works are mentioned in order to introduce the topic. In Section 3 the experimental setup is presented, the used data-set and the different experimentation carried out are explained thoroughly. To conclude, in Section 4 the obtained results are shown.

## 2 RELATED WORKS

Many different approaches have been developed to tackle the problem of automatic sign language recognition [2, 7]. In this kind of problems two main phases can be identified; the feature extraction and the classification. Through the years, mainly two ways of extracting features have been followed: using measure devices such as data gloves or body trackers, and vision based (camera based) approaches. The extracted features can refer to the whole sign or sub-units such as hand shape, hand placement, etc. On the classification phase, Hidden Markov Models and Neural Networks are widely used.

Sometimes, using just the information provided by the hands is enough to recognize the sign that is being performed. However, facial features are considered to be essential, and in most cases these features are decisive to difference similar signs. Despite its importance, few works have used facial expression analysis in combination with hand and body features. In [11], the authors propose a vision-based recognition system which extracts facial features along with hand features from input images. They use an active appearance model to identify facial areas of interest such as the eyes or mouth region, and the classification is done with a Hidden Markov Model. Their results proved the importance of facial information and they discovered that some of the gestures were correctly classified with just this facial information.

In [12] the authors propose a visual recognition system based on action recognition techniques to recognize individual words in videos of continuous ASL (American Sign Language) signing. They present a new dataset of RGB and depth videos of multi-sentence ASL, signed by fluent signers and ASL students. They use depth images along with RGB image-based facial expression, hand shapes and both body and facial landmarks as features. For each video five modalities of signals are extracted, whose features are then combined to represent the whole video. A Linear SVM (Support Vector Machine) is used for classification.

As video sequences are composed of both spatial and temporal features, in [6] the authors use the Inception model deep Convolutional Neural Network (CNN) to train on spatial features and a Long-Short Term Memory (LSTM) Recurrent Neural Network (RNN) to train on temporal features. To train the spatial features, the frames of each video are extracted and grayscale images of the hand (after removing the background and other body parts) are used to feed the CNN model. However, to train the temporal features, they propose two different training methods differing on the input of the RNN. In the first method the predictions made by the CNN are passed as input and in the second one, the last pool layer is transferred instead, which consists in a 2048 dimensional vector. The authors of [8] present a system to recognize 20 Italian gestures which uses depth information extracted by a Microsoft Kinect, Convolutional Neural Networks and GPU acceleration. They use CNNs for feature extraction and an Artificial Neural Network for classification.

The Sign Language Recognition system proposed by [4] consists of extracting human keypoints from the face, hand and body parts. They present a Korean Sign Language dataset recorded by fluent Korean signers. The keypoints are extracted using OpenPose [1] and the features vectors composed by these keypoints are then

standardized to reduce the variance of the data. For classification, stacked bidirectional Gated Recurrent Units (GRUs) are used, with 400 cells in the first layer and 200 cells in the second one.

As seen, nowadays with advanced solutions as depth cameras, wireless motion sensors and classification methods as Deep Neural Networks, this kind of tasks are becoming more feasible. However, much remains to be done.

## 3 EXPERIMENTAL SETUP

In this section, the pipeline of our approach is explained. First, the used dataset is presented, the preprocessing steps are then described and, afterwards, the classification method is explained.

### 3.1 Dataset

In the classification task presented in this work, the LSA64 dataset [10] has been used. It is an Argentinian Sign Language (LSA) dataset composed by a total of 3200 videos of 64 different LSA signs. These videos have been recorded by 10 non-expert subjects, each one repeating all the 64 signs 5 times. Within the database signs performed with both hands or just with the right hand (since all the signers are right-handed) can be found, and both nouns and verbs are included. In the videos the signs are performed using just the hands, with no particular facial expression. All the videos in the database have a resolution of  $1920 \times 1080$  and 60fps.

As a first approach a subset of the LSA64 dataset has been used, which is composed of the 42 signs that are one-handed. This way, 2100 videos compose the used database where 23 signs ( $23 \times 10 \times 5 = 1150$  videos) were recorded outdoors with natural lighting, to provide differences of illumination between signs, and 19 signs ( $19 \times 10 \times 5 = 950$  videos) were recorded indoors with artificial lighting. The signs performed by the subjects can be seen in Table 1.

Every signer wears colored gloves (red and green) in order to facilitate the task of hand segmentation, although this is not helpful in the approach presented in this paper, as no hand segmentation is performed.

### 3.2 Classification pipeline

Once the dataset is selected, the necessary features are extracted in order to perform the classification. A graphical explanation of the proposed approach can be seen in Figure 1, where the followed steps are shown. Briefly, hand information of the signers in the videos has been extracted in the form of hand skeleton joints to create a set of signals which are then used to feed the Common Spatial Patterns (CSP) algorithm. The signals are projected to another space using CSP and, from these transformed signals, the characteristics used as input for the classifiers are obtained.

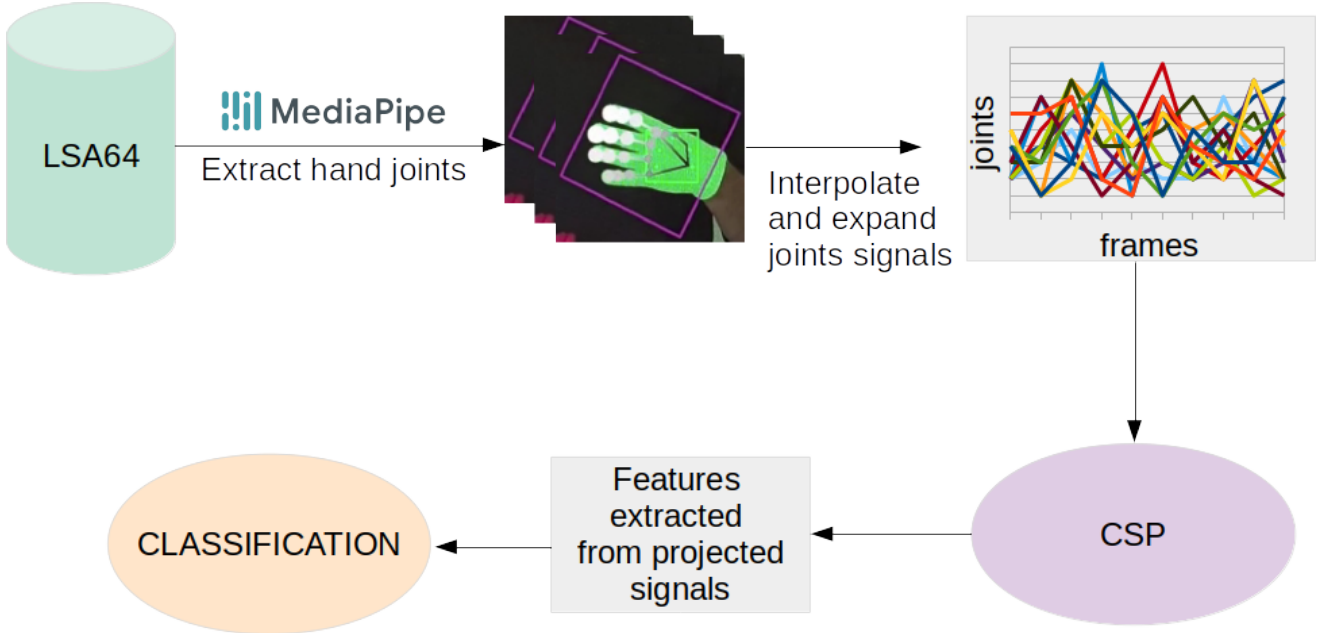
Next, each step is explained in detail.

*3.2.1 Preprocessing.* In order to get representative features from the recorded videos, MediaPipe [5] has been used, a software which offers ML solutions for streaming media. As mentioned before, in the videos used in this work the signs are performed with just the hands, with no information added by the facial expression of the signer. Therefore, only the hand landmarks have been extracted using MediaPipe Hands [13], a real-time on-device hand tracking solution. This way, the landmarks showed in Figure 2 are obtained



**Table 1: Signs used for classification, extracted from LSA64 dataset.**

Opaque	Red	Green	Yellow	Bright	Light-blue	Colors
Red2	Women	Enemy	Son	Man	Away	Drawer
Born	Learn	Call	Skimmer	Bitter	Sweet milk	Milk
Water	Food	Argentina	Uruguay	Country	Last name	Where
Birthday	Hungry	Ship	None	Name	Patience	Perfume
Deaf	Candy	Chewing-gum	Shut down	Buy	Realize	Find



**Figure 1: The pipeline followed in the presented approach.**

for the right hand in every video. To do so, the MediaPipe Hand Tracking solution has been modified in order to extract the landmarks of the videos and store them.

It has been noticed that in some of the videos MediaPipe is not able to track the hands. That might be caused by the gloves that the signers wear or the speed of the hands when the signs are performed. As a first attempt to improve the tracking of MediaPipe all the videos have been converted from RGB color space to black and white, and the results obtained with both color and black and white videos have been compared. On the one hand, in 52 color videos MediaPipe does not capture the hand, while in black and white videos this amount drops to 6. This already shows a great improvement when it comes to the tracking of the hand, and in section 4 it is discussed which type of videos are better for recognition, from the values obtained for each one.

For each landmark  $(x, y, z)$  values are obtained, where the  $z$  coordinate represents the depth of each joint in reference to the position of the wrist. These values are then used to create the group of signals for each video.

The set of signals  $S$  for  $i$ -th video is defined this way:

$$S_i^{k \times n} = \begin{pmatrix} J_{1,1} & J_{1,2} & \dots & J_{1,n} \\ J_{2,1} & J_{2,2} & \dots & J_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ J_{k,1} & J_{k,2} & \dots & J_{k,n} \end{pmatrix}$$

where  $k$  is the number of joint features,  $n$  is the number of frames and  $J_{u,v}$  is the landmark value for joint  $u$  and frame  $v$ . As each landmark is composed of  $(x, y, z)$  values, the number of joint features represented by  $k$  is 63: 3 values  $(x, y, z)$  for each one of the 21 joints  $(3 \times 21 = 63)$ .

Before applying the CSP algorithm, the signals have been interpolated and expanded. Sometimes some signal values are missing when Mediapipe is not able to detect some of the landmarks on a frame. In order to avoid these missing values and get a realistic approximation, a linear interpolation has been made. Another caveat to take into account is that the CSP algorithm needs all the input signals to have the same length. For that purpose, the maximum length has been selected (the length of the longest video) and all the signals of the other videos have been extended to that length. To expand the signals, some values obtained by a linear interpolation

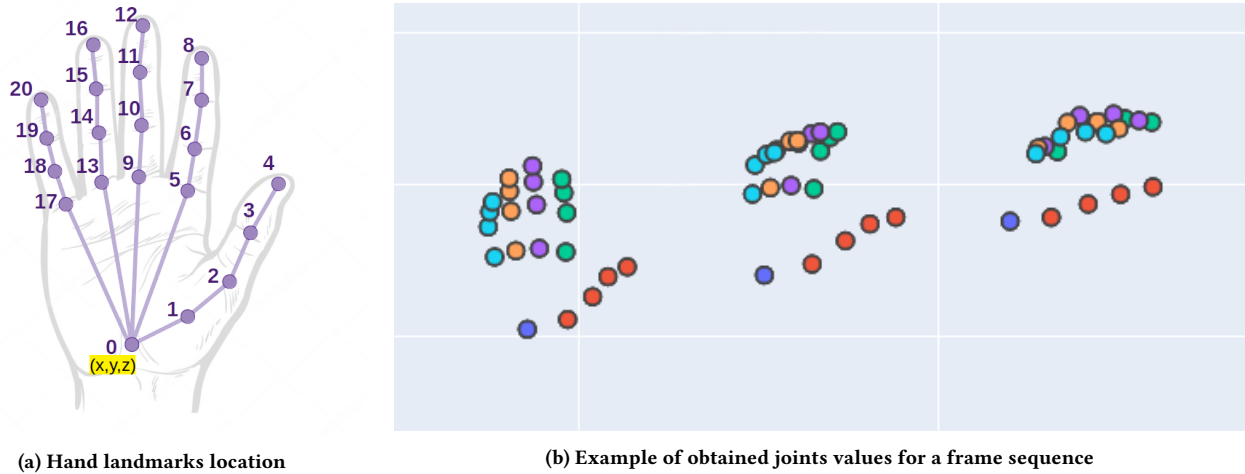


Figure 2: Hand landmarks.

are inserted between the existing ones. An example of this preprocessing can be seen in Figure 3, where the original set of signals and both the interpolated and expanded set of signals of a video are shown. In 3a the original signals are presented, which include various missing values. In 3b it can be seen how these missing values have been removed, replacing them by interpolated values. Finally, in 3c the interpolated signals are expanded to the maximum length (212 frames in this case).

As mentioned before, the signals are composed of  $(x, y, z)$  values of the joints of the right hand. However, it has been decided to also create a set of signals with just  $(x, y)$  values as the  $z$  coordinate obtained from MediaPipe is related to the wrist and this information might not be so relevant when classifying the signs, as it is not in a fixed spatial frame of reference.

**3.2.2 Common Spatial Patterns.** After creating the set of signals, the Common Spatial Patterns algorithm is executed. CSP is an extension of Principal Component Analysis (PCA) applied in signal processing and commonly used for electroencephalography (EEG) systems in Brain Computer Interface (BCI) applications.

The CSP algorithm tries to find an optimum spatial filter to reduce the dimensionality of the original signals. Considering just two different classes, those signals are projected with the CSP filter maximizing the difference of the variances between the targets. It maximizes the variance of the filtered signals of one of the classes and minimizes the variance for the other. This algorithm is usually used in EEG problems, but in this case it is used to extract characteristic features for a Sign Language Recognition problem. As explained, the signals are extracted from hand landmarks along time and these signals are the ones which are taken as input for the CSP algorithm. The CSP algorithm works with just two classes, hence all the tests are performed pairwise.

In Figure 4 an example of the projection of the signals using a CSP filter is shown. In 4a the original signals are presented, where the light-blue colored signals correspond to one class while the dark-blue colored signals correspond to the other. In order to provide a clear representation of the transformation of the signals, only 4

hand joints are shown as an example. In 4b the same signals are shown but after the transformation. It can be seen in the projected signals how the variance for light-blue class is minimized while the variance for dark-blue class is maximized.

After applying the CSP algorithm, a set of features is extracted from the projected signals in order to perform the classification. As Common Spatial Pattern filter focuses on the variances of the signals, first these variances are taken as features. When executing the CSP algorithm the value of the  $q$  variable is selected, which represents how many feature vectors are considered in the projection. The feature vectors of the spatial filter are sorted by variance, and the  $q$  first and  $q$  last vectors are selected, which produce the smallest variance for one class and the largest variance for the other class. This way,  $2 \times q$  variance values are used as features for classification.

For the purpose of performing different tests, in addition to the variances, other features are extracted from the projected signals. It has been decided to use the maximum value, the minimum value and the interquartile range (IQR) of the signals along with the previously mentioned variances to check if this extra information is useful when performing the classification.

**3.2.3 Classification.** Different classifiers have been used to perform the classification: K-Nearest Neighbors (KNN), Random Forest (RF), Naive Bayes (NB) and Support Vector Machine (SVM) with a linear kernel. As the CSP algorithm only accepts two classes as input, all the tests have been carried out pairwise.

In Table 2 the different values that the variables can take are shown. In total, 96 different configurations have been used to perform the tests, combining the values of the variables.

As the gestures in the dataset are performed by 10 different signers, it has been decided to perform a leave-one-person-out cross validation saving each time one person for testing and the rest for training, calculating the accuracy value of the model with the mean value of every test set (Figure 5). This way, it is ensured that the model is not overfitting to the people it is trained with.

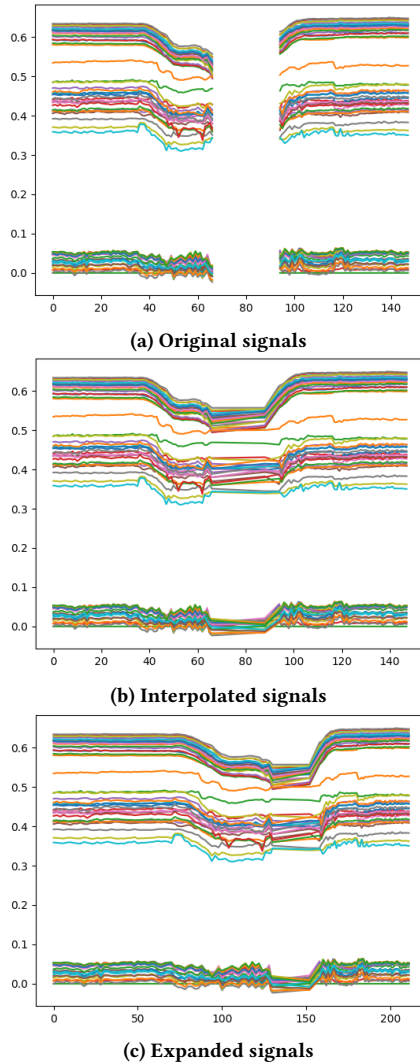


Figure 3: Preprocessing of the set of signals of a video.

Table 2: Configuration of the classification

Color space	original - black and white
Classifiers	KNN - RF - NB - SVM
q value	5 - 10 - 15
Used information	variance - variance,max,min,IQR
Used coordinates	(x,y) - (x,y,z)

## 4 EXPERIMENTAL RESULTS

In Table 3 the obtained results are shown. These results are the mean accuracy values obtained pairwise for each of the configurations.

The results show that MediaPipe works better on the black and white videos than on the original videos. It has been seen that the

joints coordinates are more accurate after original videos are converted to black and white, which could be due to the use of gloves. Regarding the coordinates, it can be concluded that  $z$  coordinate is not useful in this approach, because in general better outcomes are obtained without it. Also, considering only  $x, y$  coordinates, fewer features are used, which speeds up the classification process. Looking at the classifiers, it is clear that Naive Bayes gets the worst results, Support Vector Machine and Random Forest being the ones which fit best. When more information is used in addition to the variance, the results do not show a relevant improvement, reaching worse outcomes depending on the classifier used to perform the classification. To finish, regarding the  $q$  variable, in most cases better results are obtained when  $q = 10$ . However, the best values are achieved with different  $q$  values depending on the configuration used each time.

Looking at the results of Table 3, the best configuration is the next one:

- Dataset: B/W
- Coordinates:  $(x, y)$
- $q$  value: 10
- Classifier: RF
- Used information: variance, maximum, minimum, IQR

In order to compare the accuracy values obtained for different classes and analyze which ones are the hardest and easiest to classify, in Table 4 the mean accuracy values for each class with the best setting are shown.

The mean value for every target is  $\sim 0.98$ , obtaining high accuracy values for every pair of classes with the best configuration. It can be seen that some classes such as *Deaf*, *Candy* or *Patience* obtain almost a 100% of correct classification in each pairwise test. Other classes, for instance *Born* and *Buy*, obtain lower mean accuracy values. However, there is not a remarkable difference between the tested classes.

## 5 CONCLUSION

In this paper, a new approach for Sign Language Recognition is presented. MediaPipe has been used to extract 21 right hand joints from both original and black and white videos to create a set of signals. Then, different features have been extracted after applying CSP to perform the classification with multiple classifiers. The obtained results are promising, with an accuracy value of  $\sim 0.90$  for all configurations. The best results have been obtained with the black and white videos and a RF classifier.

In the dataset used in this work the facial expressions of the signers when they perform the signs are not relevant; they do not add any additional information to what is being transmitted by the hands. The method presented in the paper can be expanded in order to consider the facial information of the signers by adding the information that the FaceMesh solution, included in MediaPipe, obtains from the videos.

It should be mentioned that, even though in the videos used in this work the signers wore colored gloves, the method that is presented in the paper does not make use of them. This allows the method to be used with any videos where the signers do not have to use special wearables. This fact is interesting when designing a

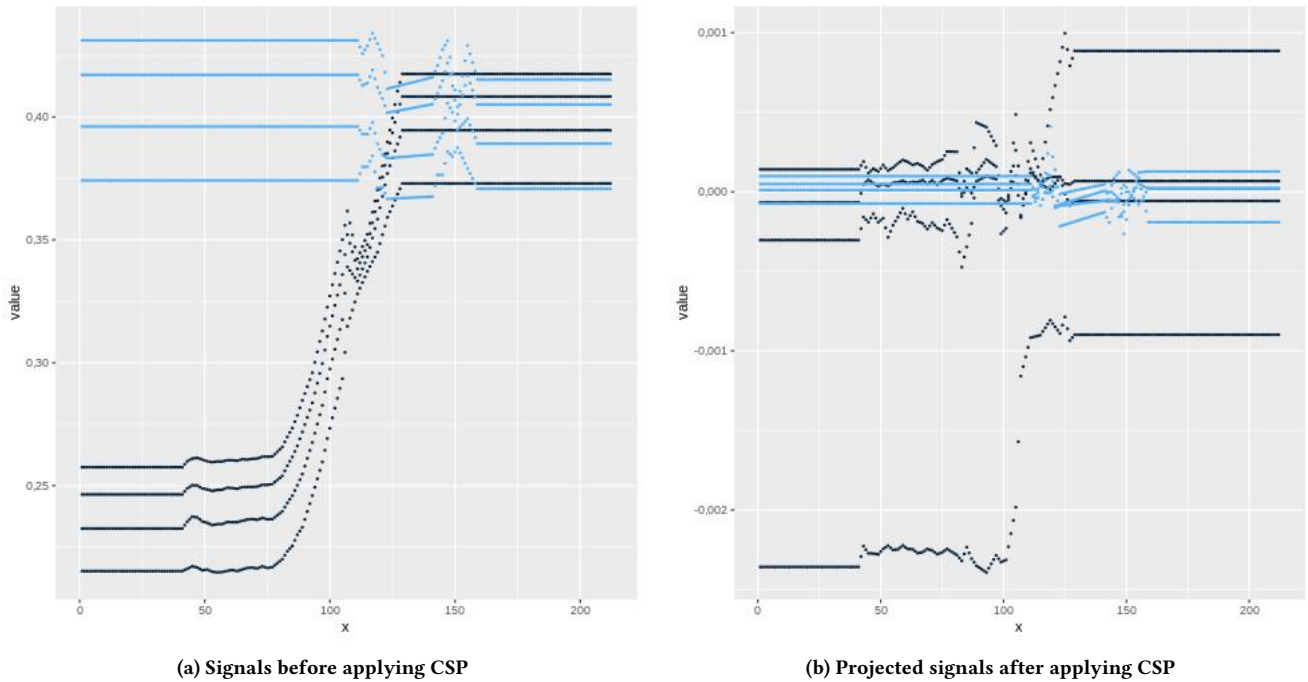


Figure 4: Signals transformation

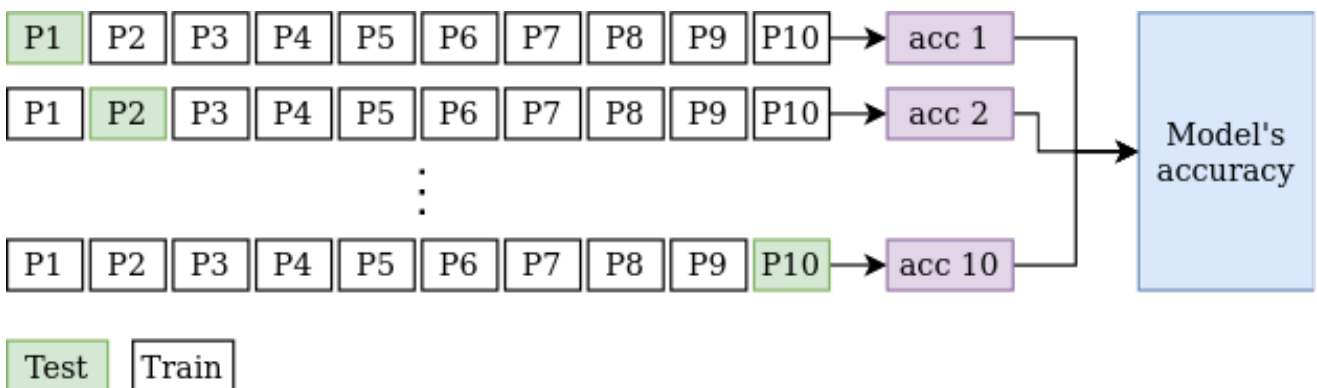


Figure 5: 10 fold cross validation with each signer of the dataset

sign recognition system to be used by people with hearing impairment, since they are not forced to wear any type of gloves or other wearables in order to be able to communicate with others.

As future work, all the classes of the LSA64 dataset could be used, adding two-handed signs which makes the classification harder. Besides, other preprocessing techniques could be applied (apart from converting images to black and white) in order to get the optimum configuration for MediaPipe to extract the correct hand joints values. Other classifiers could also be used and perform some tests with different databases.

It has been mentioned that the videos that have been used in this work have been recorded by signers that were using colored gloves. Considering that the gloves might have a negative effect on the estimation of the joint positions performed by MediaPipe, the

method should be tried with videos that do not make use of this type of wearables.

In conclusion, it has been shown that Common Spatial Patterns algorithm, usually used in processing of physiological signals, can be successfully used in other domains, i. e. Sign Language Recognition, as a feature extraction method.

### ACKNOWLEDGMENTS

This work has been partially funded by the Basque Government, Spain, grant number IT900-16, and the Spanish Ministry of Science (MCIU), the State Research Agency (AEI), the European Regional Development Fund (FEDER), grant number RTI2018-093337-B-I00 (MCIU/AEI/FEDER, UE) and the Spanish Ministry of Science,

Table 3: Obtained results with different configurations

		variance				var, max, min, IQR				
		KNN	RF	NB	SVM	KNN	RF	NB	SVM	
Color	(x,y,z)	q=5	0.9123	0.9269	0.8270	0.9404	0.8775	0.9328	0.8432	0.9396
		q=10	0.9118	0.9389	0.8789	0.9493	0.8414	0.9430	0.8836	0.9480
		q=15	0.9028	0.9363	0.8906	0.9393	0.8168	0.9379	0.8931	0.9412
	(x,y)	q=5	0.9358	0.9306	0.8697	0.9367	0.9311	0.9358	0.8785	0.9377
		q=10	0.9544	0.9457	0.9022	0.9445	0.9457	0.9475	0.9060	0.9467
		q=15	0.9546	0.9457	0.9028	0.9370	0.9456	0.9456	0.9049	0.9404
Black/white	(x,y,z)	q=5	0.9563	0.9780	0.8646	0.9801	0.9091	0.9800	0.8841	0.9801
		q=10	0.9487	0.9814	0.9074	0.9807	0.8715	0.9821	0.9152	0.9815
		q=15	0.9415	0.9778	0.9189	0.9750	0.8476	0.9787	0.9223	0.9776
	(x,y)	q=5	0.9765	0.9782	0.8980	0.9783	0.9754	0.9806	0.9113	0.9794
		q=10	0.9817	0.9807	0.9245	0.9776	0.9801	<b>0.9823</b>	0.9291	0.9795
		q=15	0.9817	0.9787	0.9219	0.9723	0.9801	0.9794	0.9249	0.9754

Table 4: Mean accuracy values obtained with the best configuration for each class

Opaque	Red	Green	Yellow	Bright	Light-blue	Colors
0.9882	0.9868	0.9805	0.9887	0.9780	0.9651	0.9658
<b>Red 2</b>	<b>Women</b>	<b>Enemy</b>	<b>Son</b>	<b>Man</b>	<b>Away</b>	<b>Drawer</b>
0.9771	0.9915	0.9837	0.9873	0.9855	0.9805	0.9867
<b>Born</b>	<b>Learn</b>	<b>Call</b>	<b>Skimmer</b>	<b>Bitter</b>	<b>Sweet-milk</b>	<b>Milk</b>
0.9695	0.9761	0.9832	0.9914	0.9868	0.9829	0.9884
<b>Water</b>	<b>Food</b>	<b>Argentina</b>	<b>Uruguay</b>	<b>Country</b>	<b>Last name</b>	<b>Where</b>
0.9723	0.9769	0.9826	0.9963	0.9776	0.9738	0.9839
<b>Birthday</b>	<b>Hungry</b>	<b>Ship</b>	<b>None</b>	<b>Name</b>	<b>Patience</b>	<b>Perfume</b>
0.9807	0.9829	0.9887	0.9831	0.9805	0.9912	0.9889
<b>Deaf</b>	<b>Candy</b>	<b>Chewing-gum</b>	<b>Shut down</b>	<b>Buy</b>	<b>Realize</b>	<b>Find</b>
0.9902	0.9907	0.9829	0.9854	0.9625	0.9703	0.9912

Innovation and Universities (FPU18/04737 predoctoral grant). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## REFERENCES

- [1] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:1812.08008* (2018).
- [2] Adil Er-Rady, R Faizi, R Oulad Haj Thami, and H Housni. 2017. Automatic sign language recognition: A survey. In *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, 1–7.
- [3] Keinosuke Fukunaga and Warren LG Koontz. 1970. Application of the Karhunen-Loève Expansion to Feature Selection and Ordering. *IEEE Transactions on computers* 4 (1970), 311–318.
- [4] Sang-Ki Ko, Jae Gi Son, and Hye-dong Jung. 2018. Sign language recognition with recurrent neural network using human keypoint detection. In *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*. 326–328.
- [5] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. MediaPipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172* (2019).
- [6] Sarfaraz Masood, Adhyan Srivastava, Harish Chandra Thuwal, and Musheer Ahmad. 2018. Real-time sign language gesture (word) recognition from video sequences using CNN and RNN. In *Intelligent Engineering Informatics*. Springer, 623–632.
- [7] Sylvie CW Ong and Surendra Ranganath. 2005. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6 (2005), 873–891.
- [8] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen. 2014. Sign language recognition using convolutional neural networks. In *Euro-pean Conference on Computer Vision*. Springer, 572–578.
- [9] Itsaso Rodríguez-Moreno, José María Martínez-Otzeta, Izaro Goienetxea, Igor Rodríguez Rodríguez, and Basilio Sierra. 2020. Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns. *Sensors* 20, 8 (2020), 2436. <https://doi.org/10.3390/s20082436>
- [10] Franco Ronchetti, Facundo Quiroga, César Armando Estrebow, Laura Cristina Lanzarini, and Alejandro Rosete. 2016. LSA64: an Argentinian sign language dataset. In *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*.
- [11] Ulrich Von Agris, Moritz Knorr, and Karl-Friedrich Kraiss. 2008. The significance of facial features for automatic sign language recognition. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, 1–6.
- [12] Chenyang Zhang, Yingli Tian, and Matt Huenerfauth. 2016. Multi-modality American sign language recognition. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2881–2885.
- [13] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv preprint arXiv:2006.10214* (2020).



# Towards an Interpretable Spanish Sign Language Recognizer

**Title:** Towards an Interpretable Spanish Sign Language Recognizer

**Authors:** I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, B. Sierra

**Conference:** The 11th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2022)

**Publisher:** SciTePress

**DOI:** 10.5220/0010870700003122

**Year:** 2022





# Towards an Interpretable Spanish Sign Language Recognizer

Itsaso Rodríguez-Moreno<sup>a</sup>, José María Martínez-Otzeta<sup>b</sup>, Izaro Goienetxea<sup>c</sup> and Basilio Sierra<sup>d</sup>

*Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU),  
Donostia-San Sebastián, Spain*

**Keywords:** Gesture Recognition, Spanish Sign Language, Interpretability.

**Abstract:** A significant part of the global population lives with hearing impairments, and the number of affected people is expected to increase in the coming decades. People with hearing problems experience daily difficulties in their interaction with non-deaf people, due to the lack of a widespread knowledge of sign languages by the general public. In this paper we present a blueprint for a sign language recognizer that takes advantage of the internal structure of the signs of the Spanish Sign Language (SSL). While the current dominant approaches are those based in deep learning and training with lot of recorded examples, we propose a system in which the signs are decomposed into constituents which are in turn recognized by a classical classifier and then assessed if their combination is congruent with a regular expression associated with a whole sign. While the deep learning with many examples approach works for every possible collection of signs, our suggestion is that we could leverage the known structure of the sign language in order to create simpler and more interpretable classifiers that could offer a good trade-off between accuracy and interpretability. This characteristic makes this approach adequate for using the system as part of a tutor or to gain insight into the inner workings of the recognizer.

## 1 INTRODUCTION

Sign languages are the main form of communication for a large proportion of people with hearing impairments. There is a great diversity of sign languages, because its evolution shares similar characteristics with spoken languages. While a significant number of non-deaf people learn non-native spoken languages out of necessity, or for professional or just intellectual reasons, deaf people tend to feel isolated even in its native communities due to the lack of interest of the general public for the sign languages.

Sign languages are quite complex, with rich grammatical structures and regional and international diversity, which makes the task of translating them into spoken languages very challenging. The signs are performed mainly with the hands, but the body position and the facial expression are also important. The hand which performs the more complex movements and moves the most is the dominant hand in the sign generation, which usually is also the dominant hand

in the everyday life of the sign speaker (left for left-handed, right for right-handed). A sign language recognizer should take into account hands, body and facial expression to perform its task correctly.

In order to favor the integration of sign language speakers, technological solutions have been devised to bridge the communication gap (Wadhawan and Kumar, 2021; Cheok et al., 2019; Er-Rady et al., 2017; Ong and Ranganath, 2005). The sign language recognition task can be divided in two main phases; the data acquisition and the classification. Regarding data acquisition there are two different approaches:

- Non-vision based, which make use of different sensors to get the information of the sign that is being performed, such us IMU (Inertial Measurement Unit) or WiFi.
- Vision based, where the acquired data are images recorded by a camera.

In addition, some of these data acquisition systems can be intrusive, for example when using data gloves, body trackers, or even colored gloves to perform hand segmentation. Depending on the captured data, different preprocessing and feature extraction methods are used (segmentation, dimensionality reduction,...). Concerning the classification, Hidden Markov Mod-

<sup>a</sup> <https://orcid.org/0000-0001-8471-9765>

<sup>b</sup> <https://orcid.org/0000-0001-5015-1315>

<sup>c</sup> <https://orcid.org/0000-0002-1959-131X>

<sup>d</sup> <https://orcid.org/0000-0001-8062-9332>

els (HMM) and Neural Networks (NN) are widely used. There is a difference between classifying static or dynamic signs; if the signs are static, a single frame has to be classified, while in dynamic signs, temporal information should also be considered.

The studies published so far have mostly focused on classifying isolated, static, one-handed signs captured by a camera and using a NN for classification, being American Sign Language (ASL) the most studied language.

In relation to Spanish Sign Language, in (Parcheta and Martínez-Hinarejos, 2017) the authors use HMMs to recognize 91 different signs captured by the Leap Motion sensor. The analyzed signs include dynamic gestures and sentences. Different HMM topologies are used, where the number of states is changed. In (Vazquez-Enriquez et al., 2021) the authors use two different architectures to perform isolated sign language recognition: a 3D Convolutional Neural Network (3D CNN) called S3D (Xie et al., 2018) for RGB data and a skeleton-based architecture called MS-G3D (Liu et al., 2020). In addition to two other datasets, they classify a subset of the LSE\_UVIGO (Docío-Fernández et al., 2020) SSL dataset. The authors of (Martínez-Martin and Morillas-Espejo, 2021) created a dataset with the Spanish alphabet which includes static and motion gestures, 18 letters and 12 letters respectively. The keypoints of the hands and arms extracted with OpenPose (Cao et al., 2019) are used to create the images which are used to perform the classification. They tried different CNN and Recurrent Neural Network (RNN) architectures to classify signs, taking into account the importance of temporal information in signs which require motion.

While many works have focused on providing some sort of feedback for spoken language learners (Pennington and Rogerson-Revell, 2019; Robertson et al., 2018), very few are dedicated to gestures in general (Banerjee et al., 2020), an even less to sign language (Paudyal et al., 2019). The aim of the system presented here is two-fold: to provide developers of machine learning models a visual way of testing and interpreting the predictions of their models, and to provide sign test students a visual and textual feedback about their performance. As a first step, only signs for which only a hand is needed are currently considered. The signs are formalized as sequences of hand configurations, where the sequence is defined as a regular expression, and the hand configurations have been learned from features derived from the spatial location of the different parts of the hand. The comparison between the intended and the recognized action is analyzed in two levels: hand configuration and sign.

The system is able to label the detected hand configuration and show the rationale of its prediction, and also the comparison with the intended sign, if they differ. With respect to the whole sign as a regular expression, where the underlying alphabet is the set of hand configurations, an explanation is also provided.

The rest of the paper is organized as follows. First, in Section 2 some basic concepts of Spanish Sign Language are explained in order to introduce the topic. In Section 3 the proposed approach is introduced, explaining the process that has been carried out. In Section 4 a discussion is presented and finally, in Section 5 the conclusions extracted from this work are presented and future work is pointed out.

## 2 SIGN LANGUAGE STRUCTURE

A sign is a combination of complex articulation positions and movements performed by a single hand (one-handed) or both hands (two-handed). In one-handed signs, the dominant or active hand is used to perform the sign. However, in two-handed signs, when the sign is symmetrical both hands act the same way, but in non-symmetrical signs the dominant hand moves while the passive hand serves as a base. Usually, the dominant or active hand is the right hand for right-handed people and the left hand for left-handed people.

Signs have four different elements (Blanco, 2009), which are equivalent to the phonemes of oral languages, and together they compose the articulation of the sign:

- Location (+ contact): the specific location where signs are performed. If a sign is performed in a corporal location, it can be in contact with that body part (+ contact) or not.
- Configuration (shape): the shape of the hand when performing the sign.
- Orientation: the orientation hands adopt when performing a sign.
- Movement: the movement usually done from the location when performing a sign.

In brief, to perform a sign, the dominant hand is placed in a location, it adopts a certain configuration and orientation in or on it, and usually performs a movement starting from that location. Nevertheless, in addition to these elements, there are some non-manual components which are fundamental to define a sign: the facial expression (eyebrows, eyes, cheeks, nose, lips, tongue) and the position of the head, shoulders or body.

As mentioned before, the shape of the hand when performing a sign is defined as a configuration. In SSL, there are three types of configurations: phonological (*queirema*), dactylogical and numerical. The phonological configurations obey a phonological system, as the distinctive sounds in oral languages, and can be classified according to different characteristics:

- Palm: extended or closed (fist).
- Fingers:
  - Extended, flexed or closed.
  - Glued or separated.
  - Which fingers are involved: index; thumb; index and thumb; middle; middle and thumb; index and middle; index, middle and thumb; pinky; pinky and index; pinky and thumb.
  - Thumb opposes the articulation of the others.

The dactylogical configurations of SSL represent the letters of the Spanish alphabet. These are used mostly when signing proper names. Lastly, the numerical configurations symbolize the natural numbers, both in isolation and incorporated in another sign.

In (Gutierrez-Sigut et al., 2016) a database of SSL is presented, where each sign is defined with the elements mentioned above, including the configurations. All the configuration and sign definitions in which this research is based have been obtained from this source.

### 3 PROPOSED APPROACH









In this section, the proposed approach and the followed pipeline are explained step by step.

**Data COLLECTION.** Although different elements as hand configuration, position, orientation or movement play a key role when recognizing a sign, as a first approach, we based the sign recognition in the recognition of different configurations and the movement from one configuration to another.

As a first approach, the eight different phonological configurations shown in Table 1 have been selected. These configurations are constituents of a wide variety of Spanish Signs as indicated in Table 1.

In the same vein, five different signs of the SSL have been chosen among the signs that use the previously selected configurations: well (*bien*), happy (*contento*), woman (*mujer*), man (*hombre*) and lis-

Table 1: Presence of selected configurations as constituents of SSL one-handed signs.

Configuration	#Signs	Configuration	#Signs
 4	124	 73	19
 50	189	 74	29
 58	55	 77	23
 59	235	 78	24

tener (*oyente*). The definitions of the mentioned signs are presented in Table 2.











A data set composed with images of the configurations which form those signs has been created. There are about 700 images for training each configuration. These values are shown in Table 3.

**Model GENERATION.** In Figure 1, the followed pipeline is shown graphically. Briefly, the method can be divided into two parts. The former is focused on the recognition of the configuration in static images, while the latter predicts the signs performed in a video using the previously trained configurations model as basis. In order to facilitate the whole process and make it easier to understand, a web app has been developed to both train new models and perform real time classification.

Since, as a first approach, it has been decided to use just the information of the hands to recognize the sign that is being performed, MediaPipe (Lugaresi et al., 2019) has been used to track the position of the hand in both images and videos. Specifically, MediaPipe Hands Tracking (Zhang et al., 2020) has been used, which offers a real-time hand tracking solution which includes 21 hand landmarks for each hand. Each hand landmark is composed of three values ( $x, y, z$ ), representing the coordinates of the key-point. In the case of the videos, these 21 landmarks are extracted for each frame.

After obtaining the landmarks for every image of the configurations data set, the features that are going to be used for training the model have to be selected. Apart from the already mentioned 21 hand-landmarks, there is the option to add the distance between finger tips or the distance from finger tips to thumb tip. These features can be selected all together,

Table 2: Definitions of the selected signs.

SIGN	INITIAL FACIAL/CORPORAL LOCATION	FINAL FACIAL/CORPORAL LOCATION	INITIAL HAND CONFIGURATION	FINAL HAND CONFIGURATION	MOVEMENT PATH
Well	Chin	High neutral space	 58	 59	Straight
Happy	Under the chin	Under the chin	 77	 78	
Woman	Right side of the neck	Under the right ear	 73	 74	Straight
Man	Close to the forehead	Close to the forehead	 4	 4	Straight
Listener	Chin	Chin	 50	 50	Circular

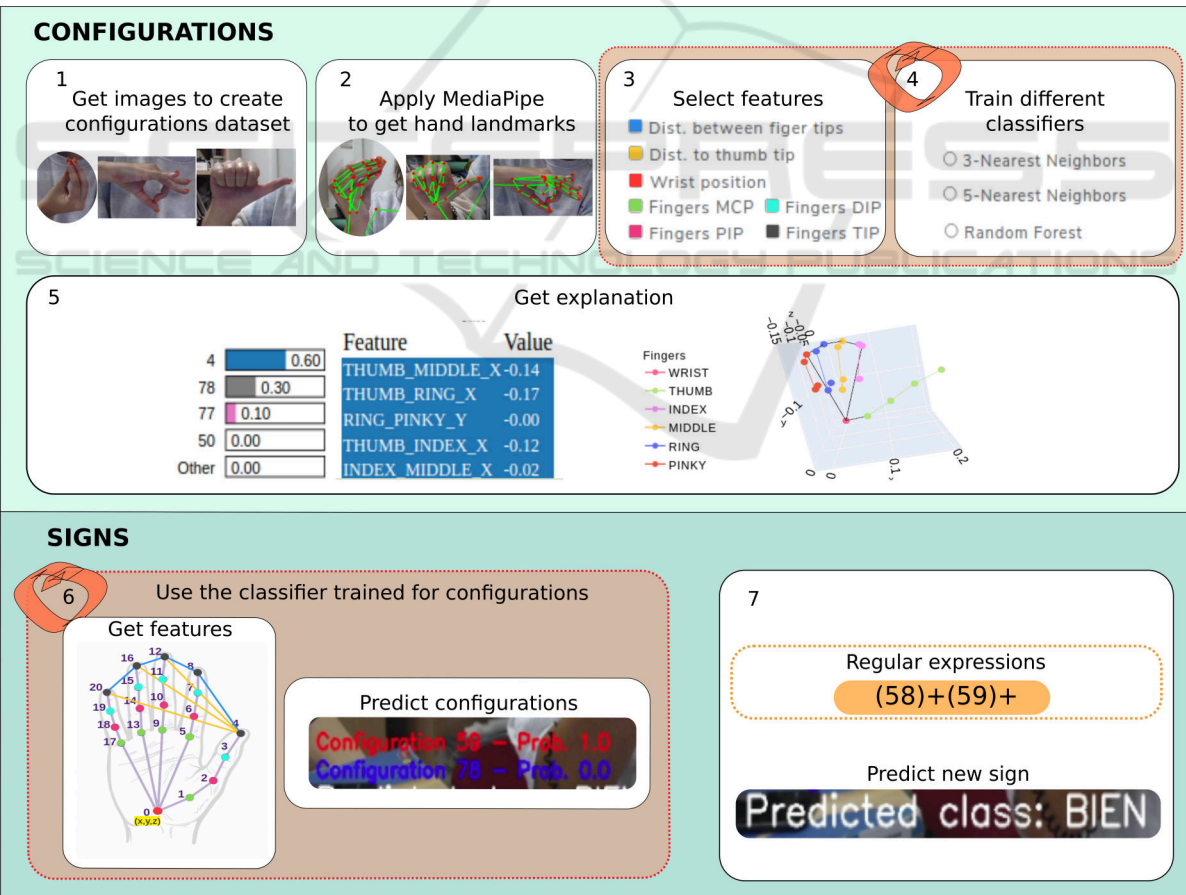


Figure 1: Pipeline. Colours in step 3 refer to positions in step 6 (MCP: Metacarpophalangeal joint; PIP: Proximal Interphalangeal joint; DIP: Distal Interphalangeal joint; TIP: Fingertip).

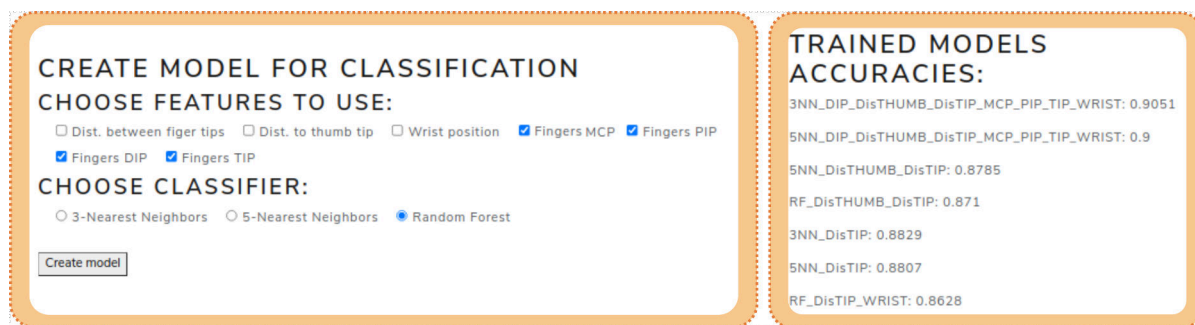


Figure 2: Training configuration model: choose features and classifier.

Table 3: Data-set.

Signs	Configurations	Number of images
Well	58	747
Man	59	804
Woman	4	700
Happy	73	732
Listener	74	743
	77	668
	78	681
	50	585

one by one or in every possible combination. Apart from that, Random Forest or K-Nearest Neighbors ( $K = 3, 5$ ) classifiers can be trained. In Figure 2 it can be seen how the training process of the configurations model is done through the web app. The accuracy values obtained for the training models are displayed aside, which can be helpful when deciding which model to use for new case predictions.

**Prediction AND EXPLANATION.** Once a model is trained, the prediction of a new image of a configuration can be done as it can be seen in Figure 3. As the goal is to develop a tutor for SSL, there is the option to choose which configuration do you want to practice. This way, an image of the configuration is shown in order to guide the user. Among all the trained models, one has to be chosen to make the new predictions. So as to decide which one to select, the accuracy values shown above give a clue of the performance of each of the trained models. If the predicted configuration corresponds to the one selected to practice, the prediction text is displayed with green background. However, if it does not match, a red background is set.

Sometimes, it is quite difficult to understand the logic behind the predictions made by a model. If an explanation of the predicted configuration is required (*Explain results* button is pressed), the two graphical items shown in Figure 4 are added, giving an explanation for a frame prediction. On the one hand, a 3D-graph is created which shows the hand landmarks



Figure 3: Real-time configuration prediction.

obtained by MediaPipe. Although the output of MediaPipe is also shown over the image the camera is recording (see top side of Figure 3), this 3D-graph mainly helps to verify if the obtained z-coordinates are correct, because they are estimated by MediaPipe from a 2D image. On the other hand, an explanation of the given prediction is obtained by LIME (Ribeiro

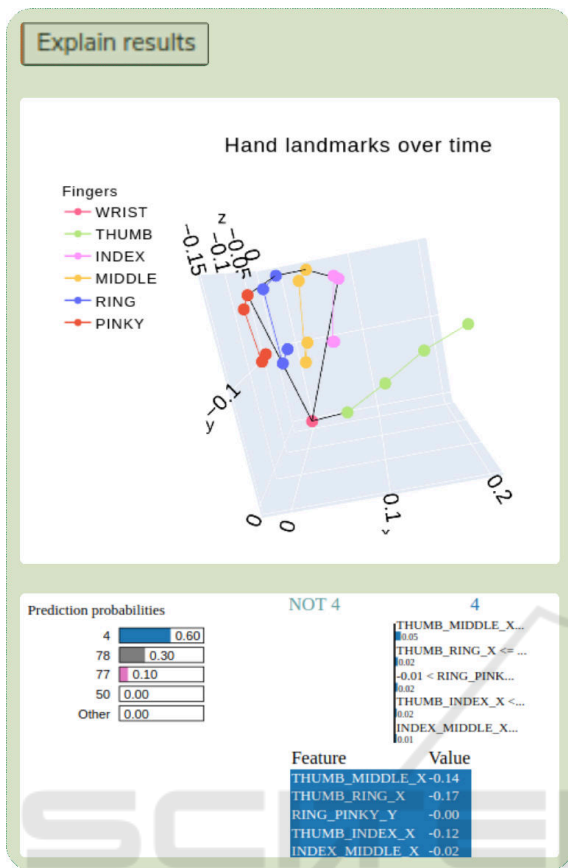


Figure 4: Explanation of the predicted configuration.

et al., 2016), a modular explanation technique which learns a local interpretable model around the prediction to give an explanation of predictions made with any classifier. As it can be seen at the bottom of the Figure 4, LIME offers several information. On the left side, the probability value of each label is indicated and, on the right side, the values of the most informative features are shown. These features might be the most informative either because they help to confirm the predicted class or because the values some of the features take indicate that the class can not be the predicted one. This way, it can be known which features have more impact when making a prediction.

Since each frame is labeled with a configuration by the classifier, a video can be summarized in a series of consecutive configuration names. Thus, a vector of configurations is obtained, a value for each frame of the video, and different regular expressions can be used to evaluate these vectors and decide which sign has been performed. The definition of the expressions can be seen in Table 4, which match with the definitions of the signs.

Using the definitions of the regular expressions, the prediction of new gestures can be performed in

Table 4: Regular expressions for each sign.

Sign	Regular expression
Well	'(58)+(59)+'
Happy	'(77)+(78)+'
Man	'(4)+(4)+'
Woman	'(73)+(74)+'
Listener	'(50)+(50)+'

real time. It has been decided to establish a sliding window of length 25 and step 1 to recognize a tentative sign, being the final prediction the mode of the last 10 predicted signs. In order to avoid the noise of incorrectly predicted configurations in between, it has been decided to establish another sliding window (within the sliding window of 25 frames) of length 10 and step 1. For each window the mode of the configurations belonging to that window is achieved, thus obtaining an array of 16 configurations ( $size\_gesture - size\_window + 1$ ) which will be the one evaluated with the regular expressions.

In the developed application, as with the configurations, it is requested to choose the gesture which is being performed to be able to indicate whether it is performed correctly or not. The models have to be chosen among the trained ones. As it can be seen in Figure 5, in addition to the predicted sign, the probability of the two most likely configurations are also indicated in order to understand the prediction. As long as a sign has not been performed (as mentioned before, the gesture length is set to 25) there is no prediction. Once a prediction can be made, a green background is established if the prediction coincides with the chosen sign and red if it does not match with the sign that was intended to reproduce.

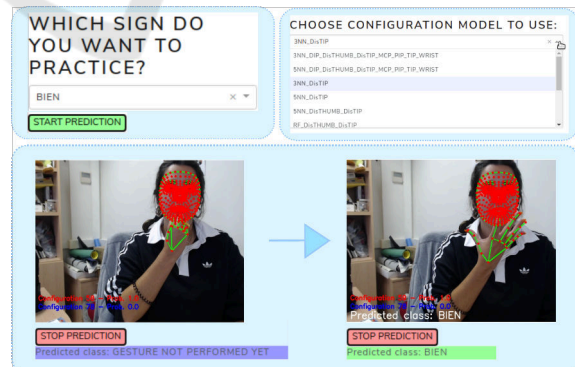


Figure 5: Real-time sign recognition.

## 4 DISCUSSION

The main goal behind the presented approach is to develop a tutor for learning Spanish Sign Language. Al-

though only the first steps are introduced, this system opens the door to many useful applications.

Since the goal is to support people who are learning sign language, improving the explanation module is crucial. The function of this module is to help to understand the results predicted by the classifier, as knowing what needs to be changed to get the desired answer can be really helpful. If the prediction is the one we expect, we can get the reason why the sign has been well performed. However, if we get an incorrect prediction, the explanation is used to indicate to the user what is being wrongly performed and, this way, the user can correct the aspects that make the sign an incorrect replica of the real sign.

This application can be approached from two different perspectives, one from the expert's side and the other from the user's side. In Table 5 the differences between both perspectives are indicated.

Table 5: Differences between the explanation given to an expert or a user.

Expert	<ul style="list-style-type: none"> <li>- <b>Knowledge:</b> the learning process of the classifier.</li> <li>- <b>Explanation:</b> LIME output.</li> <li>- <b>Action:</b> changes in the definition of the classifier.</li> </ul>
User	<ul style="list-style-type: none"> <li>- <b>Knowledge:</b> the sign.</li> <li>- <b>Explanation:</b> natural language.</li> <li>- <b>Action:</b> changes in the performance of the sign.</li> </ul>

While the expert has information about the learning process and the features that have been used to train the classification model, the user just has the visual information of the sign that he/she is learning. Hence, the information given by LIME has to be translated to natural language for the user to understand. Once the information is given, the user has the possibility to perform the sign again following the indications given by the explanation module. In the case of the expert, if the explanations received indicate that the performance of the classifier is poor (the wrong answers are due to a bad configuration of the model and not due to the performance of the user), some changes have to be done in the definition of the classification model.

As in the developed web application the sign or configuration to perform is indicated, it would be interesting if this explanation module gave information on both the chosen sign (or configuration) and the predicted one. Furthermore, although additional information apart from the hands is not considered yet, for information purposes a sentence could be added indicating the part of the body on which the sign should be performed (e.g. "Perform the sign under the chin").

## 5 CONCLUSION

In this paper the first steps towards a tutor application for learning Spanish Sign Language is presented. In the proposed approach the signs are decomposed in constituents which are in turn recognized by a classical classifier and then assessed if their combination is congruent with a regular expression associated with a whole sign. This way, unlike other systems based in deep learning, a simpler and more interpretable system is proposed, making it adequate to use for tutoring SSL and to better understand the performance of the recognizer.

As further work, we plan to extend the range of signs to recognize. Apart from the hand landmarks, specific body keypoints and the distance between them should be added as features too. Specifically in the signs used, presented in Table 2, the relevant locations are the chin, the ear and the forehead. For instance, adding the distances from the fingertips to them could be useful to distinguish between different signs. In another vein, the explanations LIME offers can be treated and displayed more clearly to the users. Taking as basis the information given for every feature, it can be translated to some sentences to inform the user what he/she should do to improve the performance of each sign (e.g. "Locate your thumb higher") as mentioned in Section 4.

## ACKNOWLEDGEMENTS

This work has been partially funded by the Basque Government, Spain, grant number IT900-16, and the Spanish Ministry of Science (MCIU), the State Research Agency (AEI), the European Regional Development Fund (FEDER), grant number RTI2018-093337-B-I00 (MCIU/AEI/FEDER, UE) and the Spanish Ministry of Science, Innovation and Universities (FPU18/04737 predoctoral grant). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## REFERENCES

- Banerjee, A., Lamrani, I., Hossain, S., Paudyal, P., and Gupta, S. K. (2020). AI enabled tutor for accessible training. In *International Conference on Artificial Intelligence in Education*, pages 29–42. Springer.
- Blanco, Á. L. H. (2009). *Gramática didáctica de la lengua de signos española (LSE)*. Sm.
- Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh,

- Y. (2019). OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186.
- Cheok, M. J., Omar, Z., and Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, 10(1):131–153.
- Docío-Fernández, L., Alba-Castro, J. L., Torres-Guijarro, S., Rodríguez-Banga, E., Rey-Area, M., Pérez-Pérez, A., Rico-Alonso, S., and Mateo, C. G. (2020). LSE\_UVIGO: A Multi-source Database for Spanish Sign Language Recognition. In *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, pages 45–52.
- Er-Rady, A., Faizi, R., Thami, R. O. H., and Housni, H. (2017). Automatic sign language recognition: A survey. In *2017 International Conference on Advanced Technologies for Signal and Image Processing (AT-SIP)*, pages 1–7. IEEE.
- Gutiérrez-Sigut, E., Costello, B., Baus, C., and Carreiras, M. (2016). LSE-sign: A lexical database for Spanish sign language. *Behavior Research Methods*, 48(1):123–137.
- Liu, Z., Zhang, H., Chen, Z., Wang, Z., and Ouyang, W. (2020). Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152.
- Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., et al. (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.
- Martínez-Martin, E. and Morillas-Espejo, F. (2021). Deep Learning Techniques for Spanish Sign Language Interpretation. *Computational Intelligence and Neuroscience*, 2021.
- Ong, S. C. and Ranganath, S. (2005). Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(06):873–891.
- Parcheta, Z. and Martínez-Hinarejos, C.-D. (2017). Sign language gesture recognition using HMM. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 419–426. Springer.
- Paudyal, P., Lee, J., Kamzin, A., Soudki, M., Banerjee, A., and Gupta, S. K. (2019). Learn2Sign: Explainable AI for Sign Language Learning. In *IUI Workshops*.
- Pennington, M. C. and Rogerson-Revell, P. (2019). Using technology for pronunciation teaching, learning, and assessment. In *English pronunciation teaching and research*, pages 235–286. Springer.
- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.
- Robertson, S., Munteanu, C., and Penn, G. (2018). Designing pronunciation learning tools: The case for interactivity against over-engineering. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13.
- Vazquez-Enriquez, M., Alba-Castro, J. L., Docío-Fernandez, L., and Rodríguez-Banga, E. (2021). Isolated Sign Language Recognition With Multi-Scale Spatial-Temporal Graph Convolutional Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3462–3471.
- Wadhawan, A. and Kumar, P. (2021). Sign language recognition systems: A decade systematic literature review. *Archives of Computational Methods in Engineering*, 28(3):785–813.
- Xie, S., Sun, C., Huang, J., Tu, Z., and Murphy, K. (2018). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321.
- Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., and Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*.



# A Hierarchical Approach for Spanish Sign Language Recognition: From Weak Classification to Robust Recognition System

**Title:** A Hierarchical Approach for Spanish Sign Language Recognition: From Weak Classification to Robust Recognition System

**Authors:** I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra

**Conference:** The 2022 Intelligent Systems Conference (IntelliSys)

**Publisher:** Springer

**DOI:** [10.1007/978-3-031-16072-1\\_3](https://doi.org/10.1007/978-3-031-16072-1_3)

**Year:** 2022





# A Hierarchical Approach for Spanish Sign Language Recognition: From Weak Classification to Robust Recognition System

Itsaso Rodríguez-Moreno<sup>(✉)</sup>, José María Martínez-Otzeta, and Basilio Sierra

Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Donostia-San Sebastián, Spain

[itsaso.rodriguez@ehu.es](mailto:itsaso.rodriguez@ehu.es)

<http://www.sc.ehu.es/ccwrobot/>

**Abstract.** Approximately 5% of the world's population has hearing impairments and this number is expected to grow in the coming years due to demographic aging and the amount of noise we are exposed to. A significant fraction of this population has to endure severe impairments even since their childhood and sign languages are an effective mean of overcoming this barrier. Although sign languages are quite widespread among the deaf community, there are still situations in which the interaction with hearing people is difficult. This paper presents the sign language recognition module from an ongoing effort to develop a real-time Spanish sign language recognition system that could also work as a tutor. The proposed approach focuses on the definitions of the signs, first performing the classification of their constituents to end up recognizing full signs. Although the performance of the classification of the constituents can be quite weak, good user-independent sign recognition results are obtained.

**Keywords:** Sign language recognition · Spanish sign language · Hidden Markov Model

## 1 Introduction

Currently about 1,500 million people live with some degree of hearing loss. Around 430 million people have a disabling hearing loss, which is equivalent to approximately 5% of the world's population. Of those affected, 32 million are children according to the World Health Organization (WHO)<sup>1</sup>. Hearing loss can be due to different causes, such as complications in childbirth, certain infectious diseases, exposure to loud sounds or ageing. Due to the continuous exposure of young people to loud noises, it is estimated that by 2050 there will be almost

<sup>1</sup> <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>.

2,500 million people with some degree of hearing loss and that at least 700 million will require rehabilitation. This suggests that the user population of Sign Language will grow. Specifically, in Spain there are more than a million people with hearing impairments and around 70,000 of them use sign languages to communicate. Sign language is not universal and there are more than 300 different sign languages around the world. Since 2007, two sign languages have been recognized in Spain: Spanish and Catalan.

Not all people who communicate in sign language have hearing impairments, and not all people with hearing impairments communicate using a sign language. Usually, sign languages are used by people with hearing impairments, family members, professionals and people who have difficulties in communicating through oral languages. However, sign languages are not yet widespread among hearing people, leading to situations where people with hearing impairments may find it difficult to communicate without an interpreter.

Therefore, in this paper an approach for automatic recognition of some signs of the Spanish Sign Language (SSL) is presented. The recognition of this language is not trivial, since, like oral languages, sign languages have their own structure, grammar and vocabulary. Thus, sign languages are visual and manual languages with their own grammar that fulfill the same functions as any other language. The presented system takes the definitions of the selected signs, extracted from [5], as the basis for the recognition task. Furthermore, we suggest a hierarchical approach where signs are recognized based on a previously trained model which classifies their constituents. We show that even with weak models such hierarchical approach can achieve good performance.

The rest of the paper is organized as follows. First, in Sect. 2 some related works are described in order to introduce the topic. In Sect. 3 the proposed approach is presented, explaining the process that has been carried out. Then, in Sect. 4 the obtained results are shown which are then discussed in Sect. 5. Finally, in Sect. 6 the conclusions extracted from this work are presented and future work is pointed out.

## 2 Related Work

In recent years, sign language recognition (SLR) has drawn increasing attention from the researchers community [4, 13, 15]. Due to the complex grammar and semantics of sign languages, their recognition and posterior translation to text is not trivial. This is mainly due to the temporal dimension of the sign, which adds difficulty in top of the extraction of relevant features in a given moment. These features could come from images recorded from cameras or from other kind of data obtained from some wearable device. Many studies have made use of data gloves to extract the features to perform the SLR. These sensors can be invasive for the signer so there is a great interest in studies that use vision-based systems to collect the data. Most SLR systems presented so far are user dependent and focus on recognizing isolated signs. A SLR system must combine pattern matching, computer vision, natural language processing and

linguistics in order to recognize the signs that are being performed and give a correct translation. These systems would be very useful in services such as hotels, stations or banks to facilitate interaction with people with hearing impairments.

Different solutions have been proposed to address this task. The authors of [11] propose a deep learning-based approach for hand SLR. First, they extract 3D hand keypoints from frames of 2D input videos using a Convolutional Neural Network (CNN) architecture and connect them to get the hand skeleton. The 3D hand skeleton is projected to three views surface images and the heatmap image of the detected keypoints is extracted. In order to obtain spatio-temporal features, a 3DCNN is applied where the pixel level information, multi-view hand skeleton and heatmap features are used as input. The obtained features are finally fed into a Long Short-Term Memory (LSTM).

Kratimenos et al. [6] extract 3D body shape, face and hands information from a single image using SMPL-X [10] model. The classification is done with a Recurrent Neural Network (RNN) consisting of one Bi-LSTM layer of 256 units and a Dense layer. They also extract skeleton information with OpenPose [2] and use these features to train the RNN in order to make a comparison, demonstrating the superiority of their approach. The use of SMPL-X holistic 3D reconstruction also obtains higher accuracy than a state-of-the-art I3D network [3] fed by raw RGB images and their optical flow.

The authors of [1] present an approach to perform Continuous Sign Language Recognition (CSLR). They introduce a Sign Language Recognition Transformer (SLR), an encoder transformer model to predict sign gloss sequences, which uses spatial embeddings of signs videos to learn spatio-temporal representations. Then, an autoregressive transformer decode model, called Sign Language Translation Transformer (SLTT), is trained to predict words and generate the corresponding spoken language sentence. In order to perform both recognition and translation, a Connectionist Temporal Classification (CTC) loss is used.

On the other hand, Ma et al. [9] propose a system for SLR using Wifi, called SignFi. In their approach, wireless signal features of sign gestures are captured collecting Channel State Information (CSI) measurements by WiFi packets. After pre-processing the raw CSI measurements to remove noises, these are used to train a 9-layer CNN to perform the classification. Specifically, the amplitude and phase of the pre-processed CSI signals are used to feed the network.

Regarding SSL recognition, the authors of [14] use the skeleton-based MS-G3D [7] architecture with the idea of retaining more reliable semantic connection between hands and body parts, as this is one important characteristic of sign languages. The MS-G3D architecture consists of stacking blocks of spatial-temporal graph convolutional networks (ST-GCN) composed by a unified spatial-temporal graph convolution module called G3D used to unify spatial and temporal features. The ST-GCN are followed by an average layer and a softmax classifier. They also use transfer learning over a SSL dataset.

In the work presented in [12] a Spanish alphabet training system is presented. A data glove, which includes an accelerometer connected to each finger, is used to acquire data. They use LabVIEW<sup>2</sup> development environment to create an

<sup>2</sup> <https://www.ni.com/es-es/shop/labview.html>.

interface for data acquisition. J48 decision tree, sequential minimal optimization (SMO) and multilayer perceptron (MLP) are used for classification. After learning the signs, the system is able to confirm if the user is performing them correctly.

The recognition of sign languages, due to the aforementioned difficulties, is a complicated task in which there is still much room for improvement.

### 3 Proposed Approach

The signs which compose the Spanish Sign Language (SSL), apart from the body position and facial expression, are defined by four main elements involving the hands:

- Hand position: The position where the hand (or hands) is located. If there are contact points (part(s) of the active hand in contact with a body part), this is also indicated. The initial position and final position might be different.
- Hand configuration: The shape of the hand. The initial configuration and final configuration might be different.
- Hand movement: The trajectory and/or movement performed by the hand.
- Hand orientation: The orientation of the palm of the hand with respect to the body of the signer. During the execution of the sign the orientation might change.

From these four elements, we propose the use of hand configurations to perform the recognition of different signs. A hierarchical approach for Spanish Sign Language recognition is presented, based on the decomposition of signs into hand configurations in order to perform the classification.

In this section, the proposed approach and the followed pipeline are explained step by step.











#### 3.1 Data Collection

As a first approach, the five different signs of the SSL presented in Table 1 have been selected: well (*bien*), happy (*contento*), woman (*mujer*), man (*hombre*) and listener (*oyente*). These signs definitions have been obtained from the SSL database presented in [5], where each sign is defined with the elements mentioned above, including the configurations and the numbers associated to them. The selected signs are one-handed and all the recorded people are right-handed.

The recognition of the signs is based on the classification of the configurations that compose these signs. For that purpose, two different data sets have been built from video sequences recorded with a webcam.

- Signs dataset: It is composed by videos corresponding to the five signs, performed by five people and with a total of 875 videos. Each video has 25 frames.
- Configurations dataset: It is a dataset composed by images of the eight configurations that are necessary to perform the selected signs. Each image refers to a configuration. Six people have been captured and the database is composed with a total of 9463 images.

**Table 1.** Definition of the selected signs and number of instances used to create the databases.

SIGN	NUMBER OF VIDEOS	INITIAL HAND CONFIGURATION	NUMBER OF IMAGES	FINAL HAND CONFIGURATION	NUMBER OF IMAGES
Well ( <i>Bien</i> )	175	 58	961	 59	1019
Happy ( <i>Contento</i> )	176	 77	875	 78	900
Man ( <i>Hombre</i> )	174	 4	915	 4	915
Woman ( <i>Mujer</i> )	175	 73	938	 74	958
Listener ( <i>Oyente</i> )	175	 50	991	 50	991
<b>TOTAL VIDEOS FOR SIGNS DATASET</b>		<b>875</b>		<b>TOTAL IMAGES FOR CONFIGURATIONS DATASET</b>	
				<b>9463</b>	

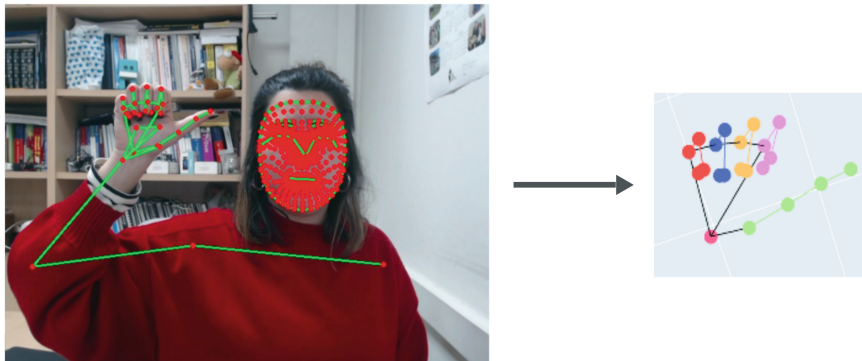
The exact numbers of instances for each sign and configuration are shown in Table 1.

In order to obtain the relevant information from the hand, since this is the body part in which we focus for this first approximation, we have used the MediaPipe Hands Tracking solution [16] from MediaPipe [8]. MediaPipe Hands Tracking, as its name indicates, performs the tracking of the hand position; more precisely, it returns twenty-one hand landmarks for each hand. Each key-point is represented by three coordinates  $(x, y, z)$ , obtaining  $21 \times 3 = 63$  values for each hand. The videos are processed frame by frame, obtaining the landmarks for every frame composing the video. An example of the solution obtained by MediaPipe algorithm is shown in Fig. 1.

As the selected signs are one-handed, and all the signers which record the database are right-handed, just the right hand information is saved. This way the created dataset shapes are  $(num\_videos, 25, 21, 3)$  for signs and  $(num\_images, 21, 3)$  for configurations, preserving the anonymity of all participants, as the original images are not recorded.

### 3.2 Followed Pipeline

The recognition is performed by decomposing the signs into constituents, and using the learned models of these constituents (the hand shape in this case) and the movement from one constituent to another to be able to classify different signs. In Fig. 2, the followed pipeline is shown graphically.



**Fig. 1.** Example of the hand landmarks extracted using MediaPipe.

Briefly, the method can be divided into two parts. The former is focused on the recognition of the configuration in static images, while the latter predicts the signs performed in a video or live feed using the previously trained configurations model as basis to train Hidden Markov Models able to recognize the signs that are being performed in real time.

As shown in Fig. 2, after applying MediaPipe, the feature selection is performed and the configurations models are trained. For videos, after selecting the features, the data are transformed using the configurations model. This way, each frame is converted to a prediction probability vector. These predictions are then used to train the HMMs which are finally used to recognize the sign that is being performed.

The details of the full process are explained next.

## Training

**Configurations.** The first part focuses on the recognition of the hand configurations, performing a static image classification. On the one hand, the position of the hand landmarks provided by MediaPipe are used as features, which include  $(x, y, z)$  values for each keypoint shown in Fig. 3a. In addition to these landmarks, it has been decided to use some additional information as feature, specifically the distances between some of the keypoints. The distances added are shown in Fig. 3b and Fig. 3c, where the distances between the thumb tip and the rest of the fingertips and the distances between contiguous fingertips are computed, respectively. The distances are independent of the spatial location of the hand and, therefore, expected to be useful when performing configuration classification.

Thus, the shape of the feature vectors is different according to the features selected. The features are handled in the three groups shown in Fig 3 (3a, 3b, 3c) and the models for all possible combinations have been trained. In Table 2 the shape of the feature vector  $f$  of instance  $i$  is indicated for each combination of features, taking into account that the feature vectors are flattened.



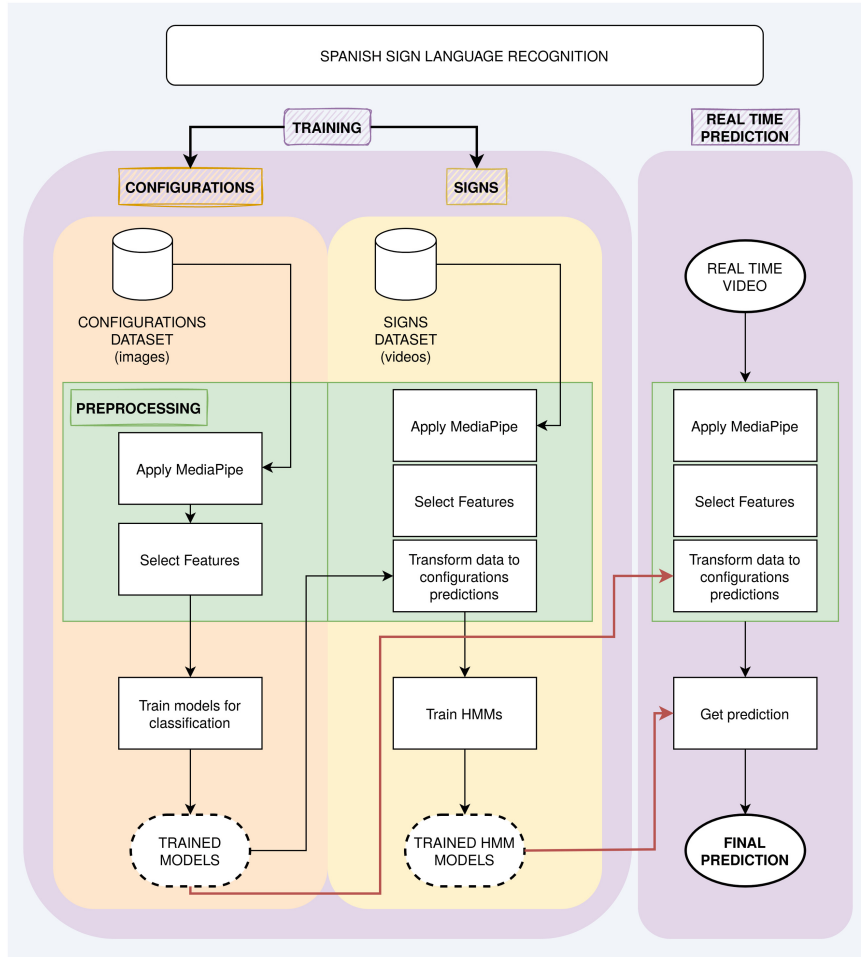


Fig. 2. Pipeline.

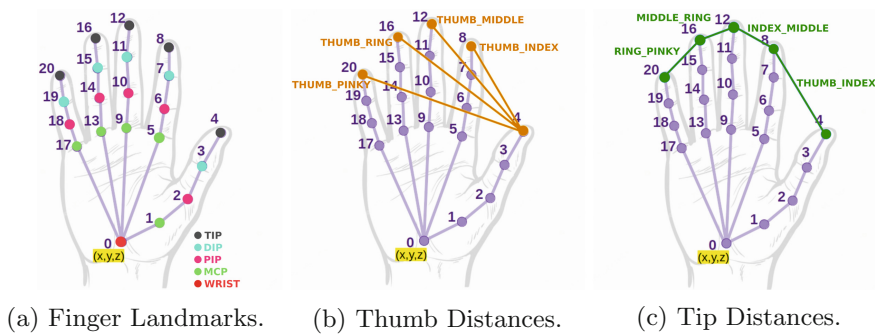


Fig. 3. Features extracted from hand joints used to train different models.

**Table 2.** Shape of feature vector  $f$  of instance  $i$  according to the selected features (\*). As THUMB\_INDEX distance is calculated in both distance types, when both are selected one is removed to avoid repeated features.

Selected features	$f_i$ shape
Hand landmarks	$(21 * 3) = (63)$
Hand landmarks + one distance	$(21 * 3 + 4) = (67)$
Hand landmarks + both distances*	$(21 * 3 + 4 + 3) = (70)$
Both distances*	$(4 + 3) = (7)$
One distance	$(4)$

Once each frame is converted to a feature vector  $f_i$ , different classifiers have been trained: Random Forest and SVM with both polynomial and Radial Basis Function (RBF) kernels. The Random Forest maximum depth is set to ten, the degree of the polynomial kernel to three and the decision function shape of the SVM to “one-vs-one”.

**Signs.** As it has been mentioned, we propose a hierarchical approach where the models trained to classify single frames containing constituents of the signs (configurations) are used to train the model which performs the classification of different SSL signs.

In order to be able to use the configuration models, the first step is to transform the signs dataset to the input format of the models. From each frame of the video the same features used for the configuration model have to be selected. For instance, if the configuration model has been trained just with the finger landmarks, the feature vector of each frame has to be formed by the values of the finger landmarks extracted for that frame, with shape  $(21 * 3) = 63$ . So far, the preprocessing of both databases is the same. However, when training the models for sign recognition another step is needed.

After representing each image in the appropriate feature space, the configuration model that we have already trained is used to get the prediction of each of the images of each video. This way, each frame is transformed to a vector of predicted probabilities, where the probability of the image corresponding to each of the eight possible configurations is indicated, as predicted by the configuration model. As we impose that all our training videos are 25 frames long, each instance  $V$  is converted to a  $25 \times 8$  matrix as the one showed in Eq. 1.  $P_{i,j}$  refers to probability  $P$  for frame  $i$  of corresponding to configuration  $j$ , which is one of the eight configurations presented in Table 1 (columns 3 and 5).

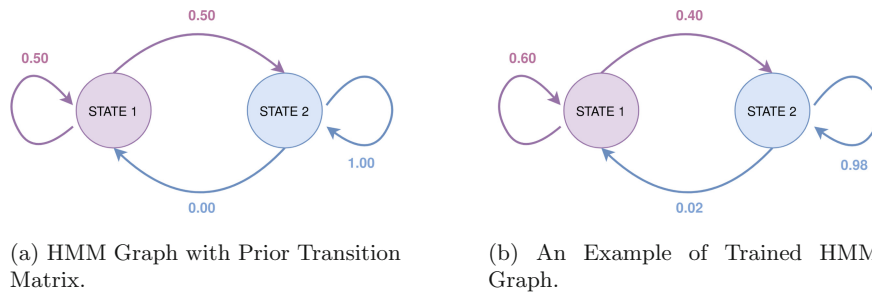
$$V = \begin{pmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,8} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,8} \\ \vdots & \vdots & \ddots & \vdots \\ P_{25,1} & P_{25,2} & \cdots & P_{25,8} \end{pmatrix} \quad (1)$$

These data, formed by the predicted probabilities of the configuration model, are used as input to train a group of Hidden Markov Models. A total of five HMMs are trained, one per sign. When training the HMM for each class, just the instances corresponding to that sign are used. According to the definition of the five chosen signs, a maximum of two configurations are used in a sign, moving from the initial configuration to the final configuration. Therefore, the HMMs are defined with two states (maximum of two configurations per sign). For example, in the case of the *well (bien)* sign, these two states would correspond to configurations 58 and 59. However, for this to happen, the models of the configurations should achieve very high accuracy. Even if this is not the case, it is intended to be able to clearly differentiate two clusters corresponding to the change from one configuration to another. The defined prior distribution of the transition matrix is shown on Eq. 2 and the prior distribution of the initial population probability on Eq. 3, considering that there is no coming back from the second state and that the signs start from the first state.

$$transmat\_prior = \begin{pmatrix} 0.5 & 0.5 \\ 0 & 1 \end{pmatrix} \quad (2)$$

$$startprob\_prior = (1 \ 0) \quad (3)$$

In Fig. 4 two HMM graph examples are show. These graphs have two states, as indicated in the definition, and the probabilities of moving from one state to the other. The graph on the left (Fig. 4a) represents the HMM before training, with the indicated number of states and the probabilities assigned in the prior transition matrix. On the other hand, the graph on the right (Fig. 4b) corresponds to a trained HMM. As it can be seen, the probabilities of moving between states have been adjusted to the data used for training. This example belongs to the class *woman (mujer)*, so the states would correspond to configurations 73 and 74 respectively, if the models used to predict configurations were perfect.



**Fig. 4.** HMM graph examples.

To classify a new video, all the frames have to be transformed to the vectors of selected features and classified by the configuration model. Once the probability

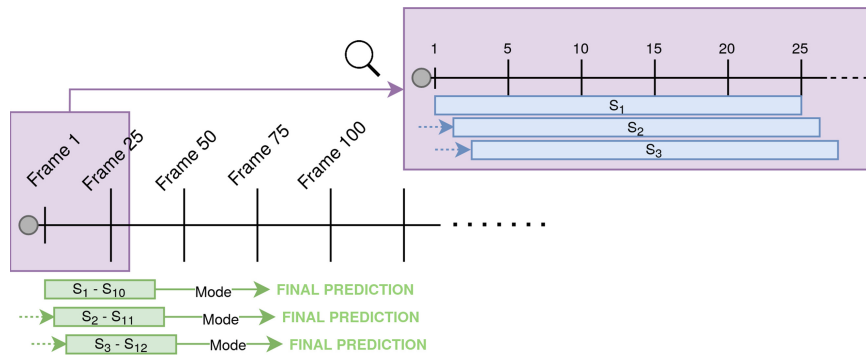
matrix (Eq. 1) is obtained, this is used as input for all the trained HMMs. The HMM with the highest score determines the class of the sign performed in the video.

**Real-Time Prediction.** After training both the configuration model and the HMMs, real-time classification can be performed. This is done in two steps: first a set of tentative signs are predicted with temporal sliding windows and then a final sign is predicted from the mode of all the tentative signs. The HMMs have been trained with videos of 25 frames length, therefore a sliding window of length 25 and step 1 is established to recognize a tentative sign. The feature vector of each frame is obtained as explained before, first the selected hand pose information is extracted and then these features are converted to a probability vector after applying the configuration classification model. When a new tentative sign is classified (25 frames compose a video), the sign corresponding to the HMM which gets a higher score for those 25 frames is predicted. Furthermore, another sliding window of length 10 and step 1 is defined to give the final prediction, which is the mode of the last 10 predicted tentative signs. In Fig. 5 a graphical representation of the real-time classification process is presented, where the mentioned sliding windows are shown.

## 4 Experimental Results

This section presents the results obtained for the trained models to get an estimation of their performance. The validation has been carried out through a Leave-One-Person-Out cross validation, using 5-folds since five different people have participated in the creation of both databases.

The notation of each of the trained models represents which features have been used to train it. In Table 3, for each name, the features used to train that model are marked.



**Fig. 5.** Real-time prediction explanation. Each  $S_i$  refers to a predicted sign. The 25 frames used to predict each tentative sign are colored in blue, while the predictions used each time to give the final prediction by calculating the mode are colored in green.

**Table 3.** Used features according to the name of each model.

Model name	Finger landmarks					Distance	
	WRIST	MCP	PIP	DIP	TIP	Thumb distances	Tip distances
DIP_MCP_PIP_TIP_WRIST	X	X	X	X	X		
DIP_DisTHUMB_DisTIP_MCP_PIP_TIP_WRIST	X	X	X	X	X	X	X
DIP_DisTIP_MCP_PIP_TIP_WRIST	X	X	X	X	X		X
DIP_DisTHUMB_MCP_PIP_TIP_WRIST	X	X	X	X	X	X	
DisTHUMB_DisTIP						X	X
DisTIP							X
DisTHUMB						X	

Following the notation presented in Table 3, in Table 4 the accuracy values obtained for each of the trained configurations models are displayed, where the best value is highlighted in bold. Twenty-one different models have been trained, using seven groups of feature vectors and three types of classifiers.

On the one hand, the best result is 0.6940 obtained SVM with RBF kernel as classifier and using finger landmarks and both distance types as features. On the other hand, the worst accuracy value is 0.5231 achieved by SVM classifier with polynomial kernel and with just tips distances as features. The range of the obtained accuracy values is not too large, but even so, certain differences are observed between the results obtained with different features and classifiers.

Regarding the feature vectors, best results are obtained using all features together. However, when only the hand landmarks are used (or in combination with any of the distances), the difference is not very notable either. The worst results are obtained using just the distances, and especially when only one of the distances is selected. Although in these cases the results are worse, it should be taken into account that the latter uses only 4 values for each instance, thus reducing the resources needed to train the models. Concerning the classifiers, in general SVM with RBF kernel obtains higher accuracy values.

**Table 4.** Configurations models accuracies.

Features	RF	SVM-poly	SVM-rbf
DIP_MCP_PIP_TIP_WRIST	0.5992	0.6734	0.6805
DIP_DisTHUMB_DisTIP_MCP_PIP_TIP_WRIST	0.6653	0.6727	<b>0.6940</b>
DIP_DisTIP_MCP_PIP_TIP_WRIST	0.6520	0.6647	0.6889
DIP_DisTHUMB_MCP_PIP_TIP_WRIST	0.6585	0.6797	0.6929
DisTHUMB_DisTIP	0.6215	0.5918	0.6292
DisTIP	0.5234	0.5231	0.5293
DisTHUMB	0.6018	0.5454	0.5969

In Table 5 the accuracy values obtained with the trained HMM models for sign classification are shown. As mentioned before, five HMMs are trained for each configuration model (one per sign). Each instance is evaluated with every HMM and the predicted output is the label of the HMM which gets the best score for the input instance. The accuracy values are calculated applying a Leave-One-Person-Out cross validation to the presented video data-set. When training the sign recognition model a previously trained configuration model is used. In order to perform the validation correctly, for each test person a configuration model trained without the instances belonging to that person is used. This way the complete evaluation is carried out on a unknown person for the model.

The best accuracy value, 0.9843 for the trained HMMs is obtained using the configuration model trained with SVM classifier with RBF kernel and hand landmarks features. The HMMs trained using the data predicted by the model trained with SVM classifier with polynomial kernel and thumbs distances features obtained the lowest accuracy value, 0.8355. Every HMM achieves better accuracy values that the underlying configuration model. There is a correspondence with the previously obtained results, as the worst values are also obtained when using the models that have performed worst when classifying configurations.

**Table 5.** Signs models accuracies.

Features	RF	SVM-poly	SVM-rbf
DIP_MCP_PIP_TIP_WRIST	0.9015	0.9398	<b>0.9843</b>
DIP_DisTHUMB_DisTIP_MCP_PIP_TIP_WRIST	0.9630	0.9198	0.9729
DIP_DisTIP_MCP_PIP_TIP_WRIST	0.9815	0.9484	0.9786
DIP_DisTHUMB_MCP_PIP_TIP_WRIST	0.9244	0.9127	0.9757
DisTHUMB_DisTIP	0.9399	0.8711	0.8483
DisTIP	0.8984	0.8625	0.9056
DisTHUMB	0.8685	0.8355	0.8469

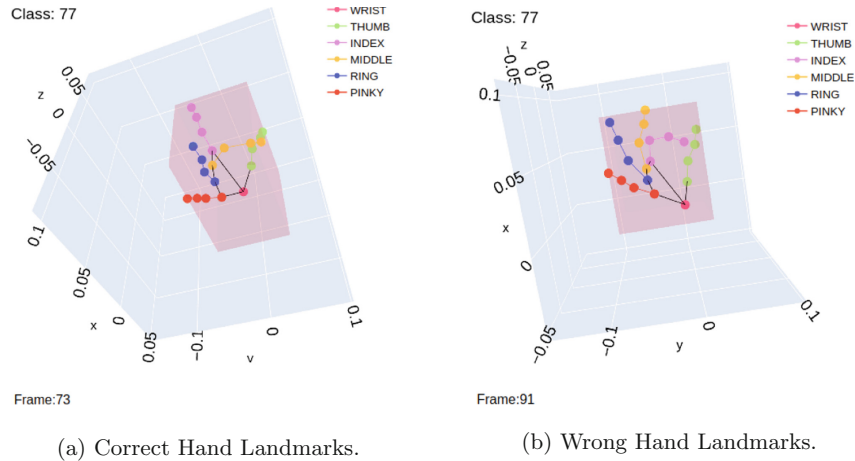
There is a significant difference between the accuracy values obtained when classifying configurations and signs. This is discussed in Sect. 5.

## 5 Discussion

In this section the obtained results are discussed, in order to shed light on them. Several difficulties are presented and analyzed in order to explain the performance of the trained models and be able to improve them in the future.

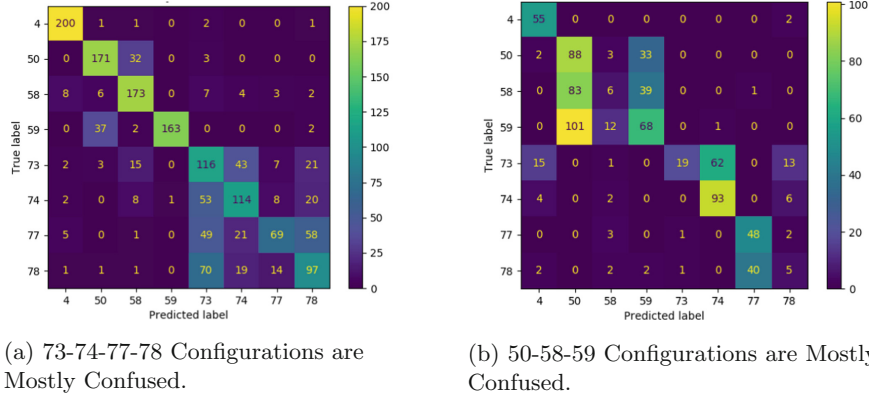
**Configurations.** As shown in Table 4, the results obtained with the models trained for the classification of configurations are not as good as might be expected. Although different factors are involved in these results, first of all the

input data has to be analyzed, the accuracy of the hand information obtained with MediaPipe. Although MediaPipe is a great technology to obtain pose estimation, it still has some weaknesses and this could lead to incorrect data collection. In Fig. 6, two examples of hand landmarks obtained with MediaPipe for configuration 77 are shown. In Fig. 6a, a correct estimation is shown, where the middle finger is flexed towards the thumb. However, in Fig. 6b, the output of MediaPipe indicates that the index finger is flexed, leading to incorrect data.



**Fig. 6.** Examples of obtained hand landmarks with MediaPipe.

Some of the configurations which form the selected signs are quite similar. For example, at first sight, 73–78 and 74–77 configurations might be misidentified. Furthermore, in the example showed in Fig. 6 of incorrect data obtained by MediaPipe, the hand landmarks of Fig. 6b which belong to class 77 could easily be considered to be an instance of class 74. In order to analyze the most misidentified classes, some confusion matrices are shown in Fig. 7. The two matrices shown correspond to two different subjects. On the left (7a), it can be seen that for this person the most misidentified classes are those already mentioned 73–74–77–78 and a square is clearly perceived where these labels are found (bottom right). On the other hand, for the person on the right (7b), although the 73–74 and 77–78 labels are also prone to confusion, mainly 50–58–59 classes are erroneously predicted. Taking into account the similarity of these classes and the possible erroneous data from MediaPipe, the accuracy values obtained for the configuration models are quite coherent.



**Fig. 7.** Examples of obtained confusion matrices.

It has been decided to perform a 10-fold cross validation over each of the people who participated in the creation of the datasets. That is, for each person only the instances corresponding to that person are used when training the model. This is done in order to analyze the degree of repeatability of each person. In conclusion, to see if the configurations are classified well when it is the same person who is performing them all the time.

**Table 6.** Configurations models accuracies: 10-Fold CV over each person.

Features	RF	SVM-poly	SVM-rbf
DIP_MCP_PIP_TIP_WRIST	0.8334	0.8326	0.8556
DIP_DisTHUMB_DisTIP_MCP_PIP_TIP_WRIST	0.8875	0.8564	0.8939
DIP_DisTIP_MCP_PIP_TIP_WRIST	0.8662	0.8414	0.8708
DIP_DisTHUMB_MCP_PIP_TIP_WRIST	0.8711	0.8517	0.8775
DisTHUMB_DisTIP	0.8212	0.8109	0.8109
DisTIP	0.7573	0.7294	0.7511
DisTHUMB	0.7605	0.7599	0.7475

In Table 6, the mean accuracy values of the trained five different models are shown. As expected, the results obtained are better than in the general case, since the classification is simpler when it is done over a single person. The results in Table 4 show that the ability of generalization of our models is limited. Still, even when single-person models are trained and evaluated, there are instances that are incorrectly predicted. To verify if the misclassified configurations coincide with the conclusions drawn previously, one of the obtained confusion matrices is shown in Fig. 8.



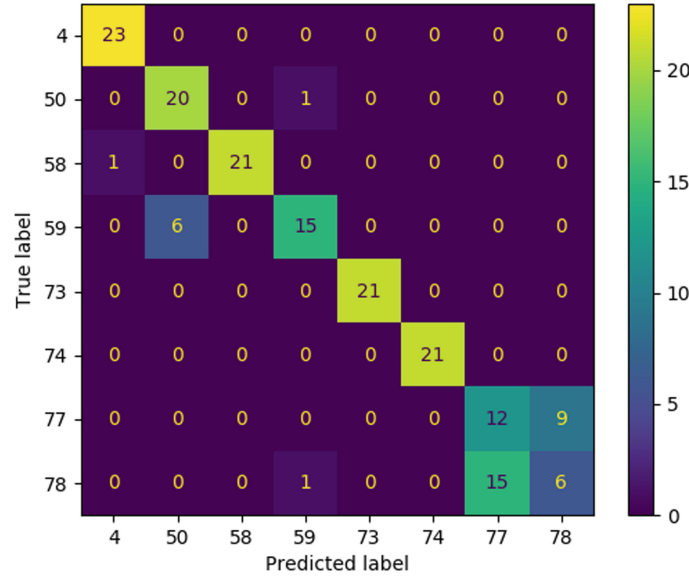


Fig. 8. Example of confusion matrix obtained using just one person of the dataset.

In this case, it is clearly perceived that the aforementioned classes 77–78 are the most misclassified configurations. So even though the training data is favorable there are clusters of configurations that are confusing, which may be due to data collection failures.

*Signs.* On the other hand, regarding the sign recognition, the classification is much better. High accuracy values are obtained for most of the trained HMMs, as presented in Table 5.

This may seem odd, since in the proposed hierarchical approach the sign recognition models use the models trained to recognize configurations as basis, and the results of these are much lower. Specifically, the predictions of configuration probabilities are used to train the HMMs. Although these predictions do not yield a high accuracy when selecting the configuration with the highest probability, they are useful for training the HMMs and recognize the sign that is being performed. Thus, the probability distribution of different people for the same sign is more similar than the probability distribution of the same person for different signs. Otherwise, the recognition would be much less efficient.

Therefore, it is concluded that a weak classification model can lead to a powerful classification model in the domain of sign language recognition, when employing a hand configuration classifier as basis for a sign classifier in a hierarchical sign recognition model.

## 6 Conclusion and Future Work

This paper presents a hierarchical approach for the recognition of some signs of the Spanish Sign Language. The selected signs are decomposed into constituents, in this case the shape of the hand (also called configuration), and the recognition of the signs is based on the classification of these constituents. To this end, different models have been trained to classify the configurations, where different features extracted by MediaPipe and several classifiers have been used. Finally, Hidden Markov Models have been used to recognize in real time a sign performed in a video or live feed. These HMMs have been trained using the predictions of the configuration models as input.

The results show that a robust recognition system can be achieved from weaker classification models. As future work we intend to analyze the source and patterns of the weaknesses of our system, therefore the estimation errors of the hand landmarks should be reduced. In addition, the use of more features has to be considered in order to make possible for our system to tell the difference between the most commonly misclassified configurations.

**Acknowledgment.** This work has been partially funded by the Basque Government, Spain, grant number IT900-16, and the Spanish Ministry of Science (MCIU), the State Research Agency (AEI), the European Regional Development Fund (FEDER), grant number RTI2018-093337-B-I00 (MCIU/AEI/FEDER, UE) and the Spanish Ministry of Science, Innovation and Universities (FPU18/04737 predoctoral grant). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## References

1. Camgoz, N.C., Koller, O., Hadfield, S., Bowden, R.: Sign language transformers: joint end-to-end sign language recognition and translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10023–10033 (2020)
2. Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., Sheikh, Y.: Openpose: realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(1), 172–186 (2019)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308 (2017)
4. Elakkiya, R.: Machine learning based sign language recognition: a review and its research frontier. *J. Ambient. Intell. Humaniz. Comput.* **12**(7), 7205–7224 (2021)
5. Gutierrez-Sigut, E., Costello, B., Baus, C., Carreiras, M.: LSE-sign: a lexical database for Spanish sign language. *Behav. Res. Methods* **48**(1), 123–137 (2016)
6. Kratimenos, A., Pavlakos, G., Maragos, P.: Independent sign language recognition with 3d body, hands, and face reconstruction. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4270–4274. IEEE (2021)

7. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 143–152 (2020)
8. Lugaresi, C., et al.: MediaPipe: a framework for building perception pipelines. arXiv preprint [arXiv:1906.08172](https://arxiv.org/abs/1906.08172) (2019)
9. Ma, Y., Zhou, G., Wang, S., Zhao, H., Jung, W.: SignFi: sign language recognition using WIFI. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 2(1), 1–21 (2018)
10. Pavlakos, G., et al.: Expressive body capture: 3D hands, face, and body from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10975–10985 (2019)
11. Rastgoo, R., Kiani, K., Escalera, S.: Hand sign language recognition using multi-view hand skeleton. *Expert Syst. Appl.* **150**, 113336 (2020)
12. González, G.S., Sánchez, J.C., Díaz, M.M.B., Ata Pérez, A.: Recognition and classification of sign language for spanish. *Computación y Sistemas* **22**(1), 271–277 (2018)
13. Sincan, O.M., Junior, J., Jacques, C.S., Escalera, S., Keles, H.Y.: Chalearn lap large scale signer independent isolated sign language recognition challenge: design, results and future research. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3472–3481 (2021)
14. Vazquez-Enriquez, M., Alba-Castro, J.L., Docio-Fernandez, L., Rodriguez-Banga, E.: Isolated sign language recognition with multi-scale spatial-temporal graph convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3462–3471 (2021)
15. Wadhawan, A., Kumar, P.: Sign language recognition systems: a decade systematic literature review. *Arch. Comput. Meth. Eng.* **28**(3), 785–813 (2021)
16. Zhang, F., et al.: Mediapipe hands: On-device real-time hand tracking. arXiv preprint [arXiv:2006.10214](https://arxiv.org/abs/2006.10214) (2020)



# Sign Language Recognition by Means of Common Spatial Patterns: An Analysis

<b>Title:</b>	Sign Language Recognition by Means of Common Spatial Patterns: An Analysis
<b>Authors:</b>	I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, B. Sierra
<b>Journal:</b>	Plos One
<b>Publisher:</b>	Public Library of Science (PLOS)
<b>DOI:</b>	10.1371/journal.pone.0276941
<b>Year:</b>	2022
<b>Source of impact:</b>	WOS (JCR)
<b>Category:</b>	MULTIDISCIPLINARY SCIENCES
<b>Impact index:</b>	3.752 (Q2)
<b>Position:</b>	29/74




## RESEARCH ARTICLE

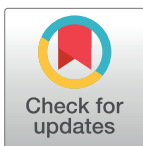
# Sign language recognition by means of common spatial patterns: An analysis

Itsaso Rodríguez-Moreno <sup>\*</sup>, José María Martínez-Otzeta , Iزارo Goienetxea , Basilio Sierra 

Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Donostia-San Sebastián, Spain

 These authors contributed equally to this work.

\* [itsaso.rodriquez@ehu.eus](mailto:itsaso.rodriquez@ehu.eus)



## Abstract

Currently there are around 466 million hard of hearing people and this amount is expected to grow in the coming years. Despite the efforts that have been made, there is a communication barrier between deaf and hard of hearing signers and non-signers in environments without an interpreter. Different approaches have been developed lately to try to deal with this issue. In this work, we present an Argentinian Sign Language (LSA) recognition system which uses hand landmarks extracted from videos of the LSA64 dataset in order to distinguish between different signs. Different features are extracted from the signals created with the hand landmarks values, which are first transformed by the Common Spatial Patterns (CSP) algorithm. CSP is a dimensionality reduction algorithm and it has been widely used for EEG systems. The features extracted from the transformed signals have been then used to feed different classifiers, such as Random Forest (RF), K-Nearest Neighbors (KNN) or Multilayer Perceptron (MLP). Several experiments have been performed from which promising results have been obtained, achieving accuracy values between 0.90 and 0.95 on a set of 42 signs.

## OPEN ACCESS

**Citation:** Rodríguez-Moreno I, Martínez-Otzeta JM, Goienetxea I, Sierra B (2022) Sign language recognition by means of common spatial patterns: An analysis. PLoS ONE 17(10): e0276941. <https://doi.org/10.1371/journal.pone.0276941>

**Editor:** Felix Albu, Valahia University of Targoviste, Universitatea Valahia din Targoviste, ROMANIA

**Received:** June 2, 2021

**Accepted:** October 17, 2022

**Published:** October 31, 2022

**Copyright:** © 2022 Rodríguez-Moreno et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** The data underlying the results presented in the study are available at <http://facundoq.github.io/datasets/lsa64/>.

**Funding:** This work has been partially funded by: - The Basque Government (<https://www.euskadi.eus/gobierno-vasco/inicio/>), Spain, grant number IT1427-22. - The Spanish Ministry of Science (MCIU) (<https://www.ciencia.gob.es/>), grant number PID2021-122402OB-C21. - The State Research Agency (AEI) (<https://www.ciencia.gob.es/portal/site/MICINN/aei/>), grant number PID2021-122402OB-C21. - The European Regional

## 1 Introduction

According to the data provided by the World Health Organization (WHO), over 5% of the world's population have some degree of hearing loss (<https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>). That sums around 466 million people (432 million adults and 34 million children), and this amount is expected to increase to around 700 million people by 2050. Among these people, more or less 70 million people (<https://wfdeaf.org/our-work/>) use one of the more than 300 sign languages that exist as first language (<https://www.un.org/en/observances/sign-languages-day>). However, as the knowledge of sign languages is not widespread around the world, these people often have difficulties to communicate in different scenarios, and their daily life interaction gets more complicated where there is no interpreter to help with the translation. In order to try to deal with these issues, many different approaches have been developed lately in the field of automatic sign language recognition.

Development Fund (FEDER)([https://ec.europa.eu/regional\\_policy/en/funding/erdf/](https://ec.europa.eu/regional_policy/en/funding/erdf/)), grant number PID2021-122402OB-C21. - The Spanish Ministry of Science, Innovation and Universities (<https://www.ciencia.gob.es/>), FPU18/04737 predoctoral grant for I. Rodríguez-Moreno. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors have declared that no competing interests exist.

Some of those approaches are kind of intrusive, requiring the signer to use some kind of wearable so the system is able to interpret what they are saying.

Sign languages, as oral languages, have their own linguistic structures and they are quite difficult to translate into spoken languages due to different aspects. Each sign language is composed of thousand of different signs which many times differ by small changes. For example, some signs have the same hand configuration but different orientation. Also, sometimes the meaning of a sign can change depending on the context or the sentence it is used in. Facial expression is also crucial to differentiate between some of the signs, which is very important for instance when making interrogative sentences. Therefore, some signs differ just in small details, such as hand configuration, movement, position, facial expression or even context.

Every sign language includes both arbitrary and iconic signs. While iconic signs are connected with what they symbolise, i.e. there is a similarity between the form of the sign and its meaning, arbitrary signs have no such connection. Iconicity [1] is noticeable both in the grammar and the lexicon of sign languages, and it can be measured in different levels [2]: *transparent* signs are easy to link with their referents, in *translucent* signs some aspects of what the signs represent are still perceived, *obscure* signs need an explanation to understand this connection, and finally, *opaque* signs have no evident relation with their referents. Other characteristics of sign languages are for example that the order of the words can be different depending on the context or that some verbs are not signed. Fingerspelling must be taken into account too, where the words are signed letter by letter. Fingerspelling is used for different purposes and its use differs in each sign language. It is mainly used for words that do not have their own sign, including proper nouns, but it can also be employed for emphasis or even for explanation when learning a sign language. Regarding the difference between sign languages, for example, in American Sign Language (ASL) fingerspelling constitutes 12%-35% of the discourse while in Italian Sign Language (LIS) it is barely used and mostly to refer to foreign words [3]. There are many other characteristics which make sign language recognition a complex task, although all of them are not mentioned here.

In this paper, an approach for video-based Sign Language Recognition (SLR) is presented. As a first step in the process, some signals are composed with the positions extracted by MediaPipe [4], which represent a set of joints of the hand which is performing the sign. These signals are then transformed using the Common Spatial Patterns [5] algorithm, a dimensionality reduction algorithm widely used in EEG signals. CSP has also been applied in the field of electrocardiography (ECG) [6], electromyography (EMG) [7, 8] or even in astronomical images for planet detection [9], and recently it has been used in video action recognition tasks [10] obtaining encouraging outcomes. This approach allows for a closed form computation and therefore it is not necessary to decide termination criteria as it happens in widely applied iterative methods, e.g., gradient descent in deep learning. The presented approach is an extension of the work introduced in [11], where the classification is performed using the feature vectors obtained after applying the CSP algorithm.

The rest of the paper is organized as follows. First, in Section 2 some related works are mentioned in order to introduce the topic. In Section 3 the experimental setup is presented, the used data-set and the different experimentation carried out are explained thoroughly. To conclude, in Section 4 the obtained results are shown and in Section 5 the conclusions extracted from this work are mentioned.

## 2 Related works

As mentioned above, sign languages have complex grammatical structures, and a sign language recognition system should involve both sign language linguistics and gesture



recognition. Sign language recognition can be divided in two different tasks; word-level recognition, which involves the recognition of isolated signs, and sentence-level recognition, where the aim is to recognize continuous signs. Due to the aspects mentioned before, both tasks are challenging.

Several sign language recognition approaches have been developed in the last years [12–14] which consist of three main phases: feature extraction, temporal-dependency modeling and classification. As previously mentioned, even though hand movements and facial expression are both important to interpret the signed language, few approaches use facial expression information [15, 16].

The methods for extracting hand features can be divided into intrusive and non-intrusive categories. While in intrusive systems there is a need to interfere with the signer to perform the feature extraction, for example with the use of colored or electronic gloves, in non-intrusive systems vision-based recognition approaches are used, where there is no need of using wearables and features extracted from RGB and depth images are used to perform the classification. Regarding the data used for classification, most of the studies make use of manual features, such as hand location, motion, configuration and orientation. Research in optimized feature extraction has also been done, e.g. using genetic algorithms [17].

Several examples of intrusive systems have been developed. Rosero-Montalvo et al. [18] present an electronic glove system to perform the SLR. The glove is composed of five flex sensors (one in each finger) and an Arduino Lilypad which reads the sensors. K-Nearest Neighbors (KNN) is used for classification. In [19] the authors developed a data glove customized with angle sensors at the finger joints and wrist. The data obtained from these sensors are directly converted into digital with a controller unit and for the recognition they use a Radial Basis Function kernel Support Vector Machine (RBF-kernel SVM).

Through the years, two different types of non-intrusive systems have been used for feature extraction for sign language recognition: sensor-based systems and vision-based systems. Different types of sensors have been used to obtain the information related to the body part positions of the signer.

In [20], the authors use the Channel State Information (CSI) of each sign gesture measured by WiFi packets as feature for their recognition system. After processing the signals to remove noise, a 9-layer CNN is fed to perform the classification. In the approach presented in [21], two depth sensors located at different viewing angles are used to capture 3D gestures, Leap Motion and Microsoft Kinect. After obtaining the positions of the fingerprints from the data acquired with both sensors, different fusion techniques are used to perform the gesture recognition; early fusion, late fusion and coupling fusion with Coupled Hidden Markov Model (CHMM). In a related research [22] the same authors use HMM, Bidirectional Long Short-Term Memory Neural Network (BLSTM-NN) and their combination for the recognition.

On the other hand, lately more approaches are being developed which are based on vision. In the approach presented in [23] first a hand segmentation is performed using a dynamic skin detector based on the color of the face. The hands are identified with the segmented skin blobs and their tracking is performed using the head as a reference point to define the hands. The coordinates of the center of the hands, the velocity of the hand movement and the orientation of the main axis of the hand are then used to compose the feature vectors, which are classified using the Euclidean distance. Pu et al. [24] propose an architecture which includes a 3D Residual Network (3D-ResNet) to extract features from input videos and an encoder-decoder network for sequence modelling, where a Bidirectional Long Short-Term Memory (BLSTM) encoder and both a Long Short-Term Memory (LSTM) decoder and a connectionist temporal

classification (CTC) decoder are used. In [25, 26] CNNs are used to perform the SLR. The authors of [27] use OpenPose [28] to extract 2D skeleton data of the body, hands and face from RGB videos, and project them to the 3D space using a deep multi-layer neural network. They also add CNN-based mouth and hands regions-of-interest and employ an encoder-decoder for recognition. In a research related to the more general human-computer interaction area [29], the authors apply crow search algorithm (CSA) [30] to select optimal hyper-parameters for CNNs trained to deal with hand gesture classification. They achieve perfect training and test accuracy over their data.

The small size of the majority of available sign language databases makes it difficult to train models that can generalize well in practice. To try to alleviate this in [31] the authors make publicly available a large-scale Word-Level American Sign Language (WLASL) video dataset, containing more than 2000 words performed by over 100 signers. They also propose a novel pose-based temporal graph convolution networks (Pose-TGCN) that models spatial and temporal dependencies in human pose trajectories simultaneously, achieving good performances, with up to 66% for the top-10 accuracy metric. Another large dataset, How2Sign, with more than 80 hours of continuous American Sign Language videos along with transcriptions, speech recordings and depth information is presented in [32]. They also create, from that dataset, synthetic videos that can be understood by ASL signers, according to a study which they also present in the paper.

Some conferences host challenges where several teams compete to best perform a task over a given dataset. In [33] the authors present the main results of the ChaLearn LAP Large Scale Signer Independent Isolated SLR Challenge, organised at CVPR 2021. Participants in two tracks (RGB and RGB+Depth) had to recognise 226 types of signs from a Turkish Sign Language dataset with 36,302 video by 43 signers. The winning entries achieved accuracy figures above 96%, with approaches combining body part estimation, external data, transfer learning, ensemble models, data fusion and spatio-temporal feature extraction. However, even the best methods still face difficulties to tell apart very similar signs, in particular when the signing hand movements are similar.

Related to sign classification, but with their own challenges, another two research fields are worth mentioning: *sign spotting* and *sign language translation*. In *sign spotting* the task is to identify the starting and ending temporal moments of a sign in a video of continuous sign language. Usually it is also possible that no sign is present in the segment video to analyze. An approach integrating learning from sparsely labelled footage, subtitles and visual sign language dictionaries is presented in [34], where these three information sources are integrated into a unified learning framework guided by noise contrastive estimation and multiple instance learning. A validation of this approach on low-shot sign spotting benchmarks is also presented. In *sign language translation* the goal is to generate natural language sentences in text representation from a sequence of sign language video. In [35] a temporal semantic pyramid network, called TSPNet, is introduced, with inter-scale and intra-scale attention to achieve local semantic consistency as well as solving ambiguity using non-local information. The authors test their method on the RWTH-PHOENIX-Weather 2014T (RPWT) dataset [36] and claim to improve the performance of state of the art methods according to the BLEU and ROUGE scores.

In [Table 1](#) an overview of the approaches mentioned in this section for sign classification is displayed for a better understanding.

The advances in depth cameras, wireless motion sensors and classification methods as Deep Neural Networks, are making the sign language recognition task more feasible. However, due to the difficulties mentioned above, such as the scarcity of large databases or the complexity of the sign languages, much remains to be done.

Table 1. Overview of the mentioned approaches.

	Data Collection Technique	Classification Method	Dataset
[18]	Electronic glove (flex sensors + Arduino)	KNN	Numbers 1-10
[19]	Data glove (accelerometer)	SVM (RBF-kernel)	American SL alphabet Indian SL alphabet (one-handed) + numbers
[20]	WiFi packets	CNN	American SL 276 signs
[21]	Leap motion Microsoft Kinect	Coupled HMM	Indian SL 25 dynamic signs
[22]	Leap motion Microsoft Kinect	HMM + BLSTM	Indian SL 50 dynamic signs
[23]	Hand segmentation (skin detector)	Euclidean distance	Arabic SL 30 isolated words
[24]	Video representation (3D-ResNet)	BLSTM encoder LSTM and CTC decoder	RWTH-PHOENIX-Weather German SL dataset CSL dataset with 178 Chinese words
[25]	Video frames	CNN	ISL 200 words
[26]	Video frames	CNN + SVM	American SL alphabet + numbers
[27]	Estimated 3D hand poses (2D hand skeleton Openpose + Neural Network)	Attentional CNN encoder-decoder	Greek SL 306 isolated words ChicagoFSWild dataset
[31]	Video frames	Pose-based Temporal GCN	WLASL 2000 words
[33]	Video frames + depth	Multiple methods	AUTSL (Turkish Sign Language) 226 signs

SL: Sign Language.

<https://doi.org/10.1371/journal.pone.0276941.t001>

### 3 Experimental setup

In this section, the pipeline of our approach is explained. First, the used dataset is presented, the preprocessing steps are then described and, afterwards, the classification method is explained.

#### 3.1 Dataset

Although there are some databases with more than a thousand classes [36–38], most of the current datasets are not very large [39–41]. In this case, an Argentinian Sign Language (LSA) dataset, LSA64 dataset [42] is used, which is composed of 64 different LSA signs. There are 3200 videos in total, with each sign begin repeated five times by 10 non-expert subjects. Both one-handed (42 signs performed with the right hand) and two-handed (22 signs) signs can be found. The subjects wore black clothes and colored gloves (red and green), being recorded with a white wall as background in an indoor and an outdoor environment. The colored gloves (red and green) are used in order to facilitate the task of hand segmentation, although this is not helpful in the approach presented in this paper, as no hand segmentation is performed. When performing the signs, the subjects do not make use of the facial expression, they just focus on the movements of the hands. All the videos have a resolution of  $1920 \times 1080$ , 60fps and have been recorded placing the camera 2m away from the wall.

In order to simplify the classification problem, as a first approach a subset of the dataset has been selected, precisely the 42 one-handed videos have been used. The name and information of the used signs can be seen in Table 2. Thus, the subset used is composed by 2100 videos, where 1150 videos were recorded outdoors with natural lighting (23 signs, 10 signers, 5 repetitions) and 950 videos were recorded indoors with artificial lighting (19 signs, 10 signers, 5 repetitions).

#### 3.2 Classification pipeline

The proposed approach's pipeline is shown in Fig 1, where three main phases can be distinguished: data acquisition, feature extraction and classification. Briefly, in the data acquisition

Table 2. Signs used for classification, extracted from LSA64 dataset.

CLASS	ID	ENV.	CLASS	ID	ENV.	CLASS	ID	ENV.
<i>Opaque</i>	001	Indoor	<i>Born</i>	015	Indoor	<i>Birthday</i>	030	Outdoor
<i>Red</i>	002	Indoor	<i>Learn</i>	016	Indoor	<i>Hungry</i>	033	Outdoor
<i>Green</i>	003	Indoor	<i>Call</i>	017	Indoor	<i>Ship</i>	037	Outdoor
<i>Yellow</i>	004	Indoor	<i>Skimmer</i>	018	Indoor	<i>None</i>	038	Outdoor
<i>Bright</i>	005	Indoor	<i>Bitter</i>	019	Indoor	<i>Name</i>	039	Outdoor
<i>Light-blue</i>	006	Indoor	<i>Sweet milk</i>	020	Indoor	<i>Patience</i>	040	Outdoor
<i>Colors</i>	007	Indoor	<i>Milk</i>	021	Indoor	<i>Perfume</i>	041	Outdoor
<i>Red2</i>	008	Indoor	<i>Water</i>	022	Indoor	<i>Deaf</i>	042	Outdoor
<i>Women</i>	009	Indoor	<i>Food</i>	023	Indoor	<i>Candy</i>	046	Outdoor
<i>Enemy</i>	010	Indoor	<i>Argentina</i>	024	Outdoor	<i>Chewing-gum</i>	047	Outdoor
<i>Son</i>	011	Indoor	<i>Uruguay</i>	025	Outdoor	<i>Shut down</i>	052	Outdoor
<i>Man</i>	012	Indoor	<i>Country</i>	026	Outdoor	<i>Buy</i>	059	Outdoor
<i>Away</i>	013	Indoor	<i>Last name</i>	027	Outdoor	<i>Realize</i>	062	Outdoor
<i>Drawer</i>	014	Indoor	<i>Where</i>	028	Outdoor	<i>Find</i>	064	Outdoor

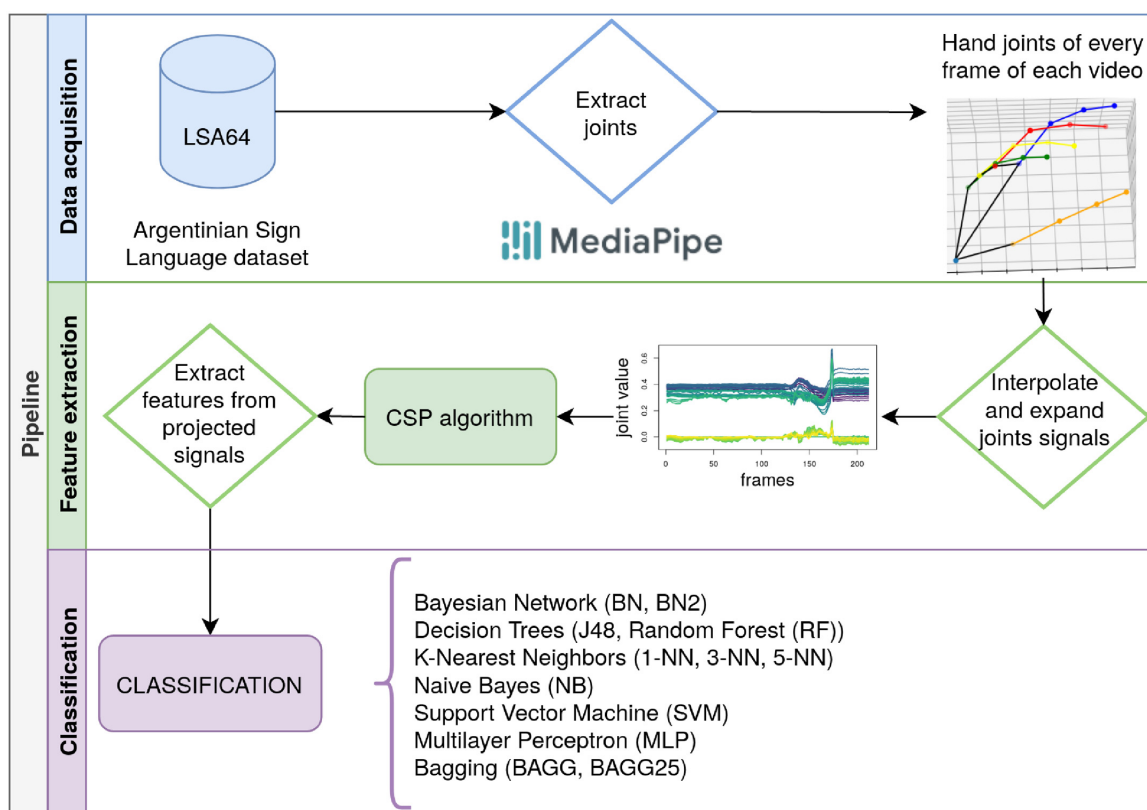
<https://doi.org/10.1371/journal.pone.0276941.t002>

phase, the desired information is extracted from the original videos of the database. In this case, after selecting the dataset, the hand landmarks positions are obtained. Then, in the feature extraction phase, these hand landmarks are processed and a set of features is obtained after applying the Common Spatial Patterns algorithm. To finish, the classification is performed using different classifiers to make a comparison between them. The following subsections contain a detailed explanation of each stage.

**3.2.1 Data acquisition.** Since in the videos of the selected dataset the signers only use their hands to perform the signs and their facial expression should not be taken into account, it has been decided to track the positions of the hands in each frame of the video. For that purpose, a technology called MediaPipe [4] has been used, more specifically the MediaPipe Hands Tracking [43] solution. This provides a real-time hand tracking solution which includes the hand landmarks showed in Fig 2 for both hands. For our approach, we have queried the MediaPipe Hand Tracking solution API for the right hand landmarks for every frame of the videos and stored them. Each landmark is composed of the three coordinates (x, y, z) which denote its spatial location. The z coordinate represents the depth of each joint in reference to the position of the wrist.

Once the landmark values are obtained, a set of signals is created for every video of the database. The coordinate values of the joints are used to create the group of signals S for each video i, which is defined this way:

$$S_i^{3k \times n} = \begin{pmatrix} J_{1,x,1} & J_{1,x,2} & \cdots & J_{1,x,n} \\ J_{1,y,1} & J_{1,y,2} & \cdots & J_{1,y,n} \\ J_{1,z,1} & J_{1,z,2} & \cdots & J_{1,z,n} \\ J_{2,x,1} & J_{2,x,2} & \cdots & J_{2,x,n} \\ \vdots & \vdots & \ddots & \vdots \\ J_{k,z,1} & J_{k,z,2} & \cdots & J_{k,z,n} \end{pmatrix}$$



**Fig 1. The pipeline followed in the presented approach.**

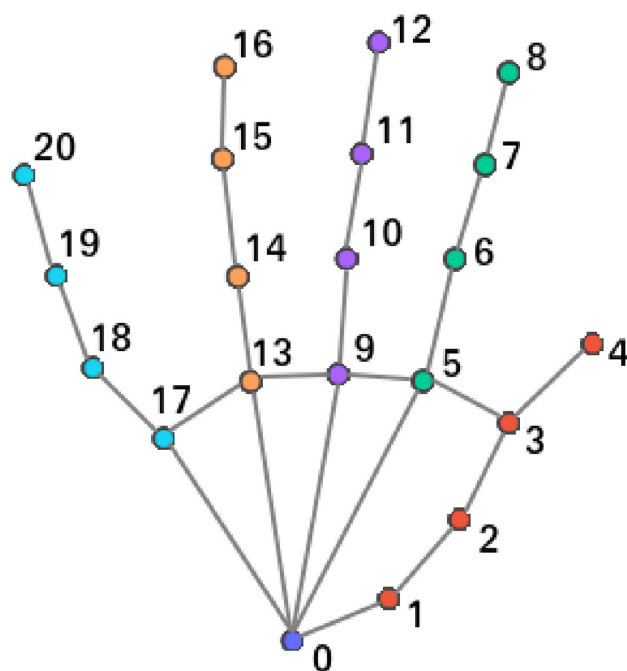
<https://doi.org/10.1371/journal.pone.0276941.g001>

where  $k$  is the number of joint features,  $n$  is the number of frames and  $J_{u,c,v}$  is the landmark value for joint  $u$ , coordinate  $c$ :  $x$ ,  $y$ ,  $z$  and frame  $v$ . The number of joints extracted for each frame is 21 ( $k = 21$ ), and as each landmark is composed of  $(x, y, z)$  values, the number of rows of the signal matrix is 63: 3 values ( $x, y, z$ ) for each one of the 21 joints ( $3 \times 21 = 63$ ). As the  $z$  coordinate is related to the wrist might be irrelevant when performing the classification. To test this hypothesis, it has been decided to also perform the classification with just  $(x, y)$  coordinates, creating a signal matrix of 42 rows: 2 values ( $x, y$ ) for each one of the 21 joints ( $2 \times 21 = 42$ ).

In Fig 3 an example of a sequence of a hand performing a sign can be seen, where the hand landmarks obtained by MediaPipe are shown graphically.

It has been observed that in 52 of the dataset's original videos, MediaPipe does not track the hand throughout the entire video. This may be due to the speed of the movement of the hands when performing the signs or the use of the color gloves worn by the signers, which can hinder the application of MediaPipe. It has been decided to convert the original videos from RGB color space to black and white in order to try to improve the tracking of MediaPipe. Using the black and white videos, the number of videos where the hand is not detected in any frame of the video drops from 52 to 6. Thus, it can be concluded that applying some preprocessing to the original videos the performance of MediaPipe can be improved.

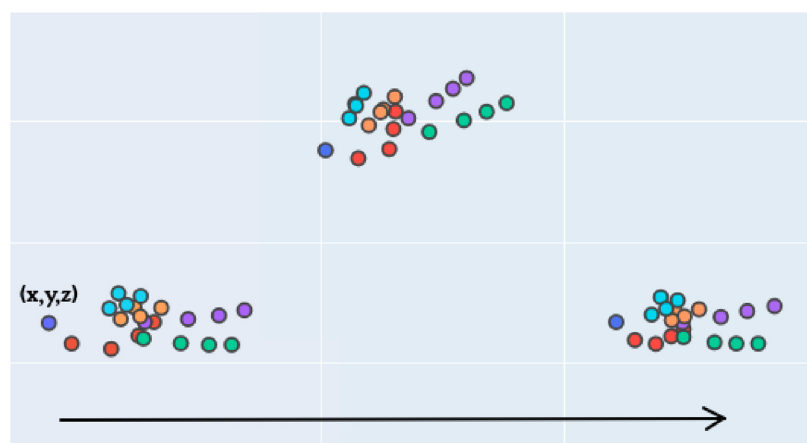
**3.2.2 Feature extraction.** In the second phase, the features for the classification are extracted from the signals created with the landmarks obtained by MediaPipe.



**Fig 2. Hand landmarks obtained with MediaPipe.**

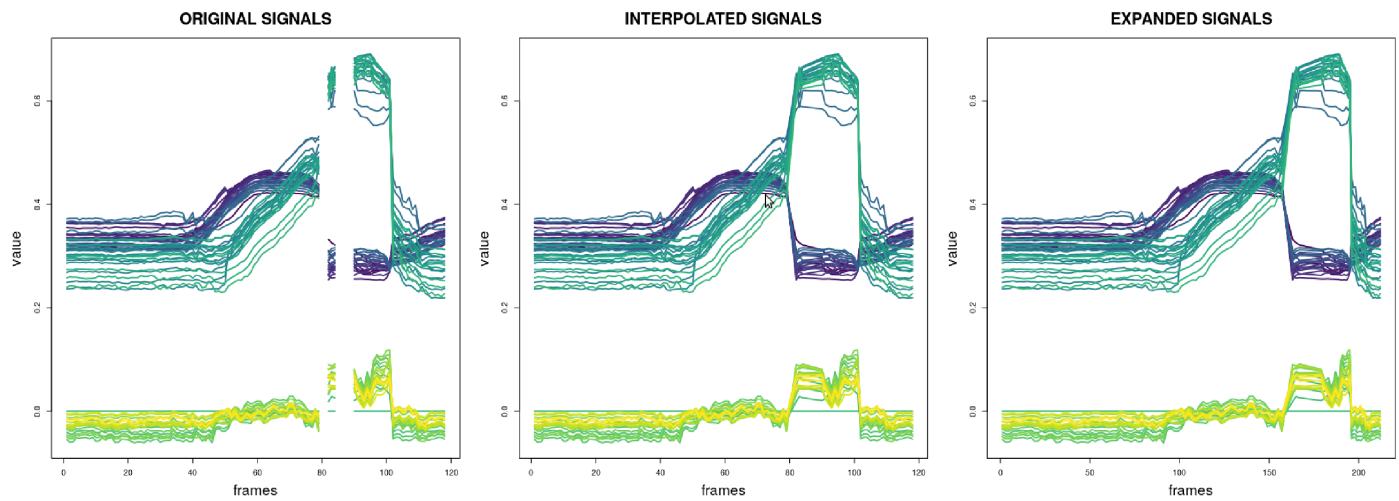
<https://doi.org/10.1371/journal.pone.0276941.g002>

First of all, interpolation is used to fill in the missing values in the signals. Sometimes MediaPipe is not able to capture any or some of the landmarks on the frame that is being processed, leading to a set of signals with missing values. A linear interpolation is performed to replace these missing values, trying to get a realistic approximation. Once the signals are completed and having removed all the missing values, the input signals have been extended to the same length because the Common Spatial Patterns algorithm needs all the input signals to have the same length. This way, the maximum length has been selected (the length of the longest video) and all the signals have been expanded to that maximum length, inserting some new values obtained by a linear interpolation between the existing ones. In Fig 4 an example of the



**Fig 3. Example of hand landmarks obtained for a sign sequence.**

<https://doi.org/10.1371/journal.pone.0276941.g003>



**Fig 4. Preprocessing of the set of signals of a video.**

<https://doi.org/10.1371/journal.pone.0276941.g004>

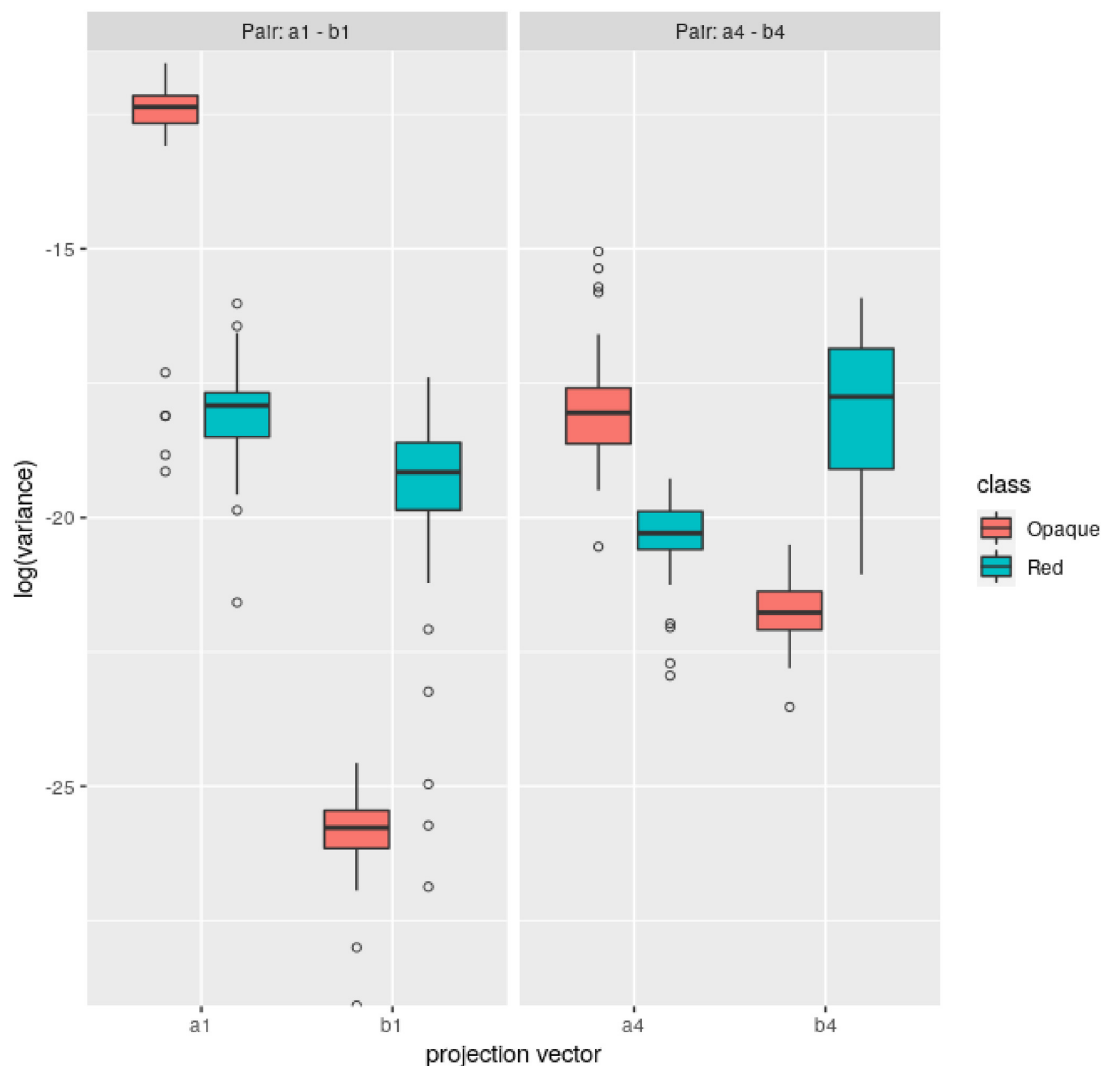
explained interpolation and expansion of the signals is shown. It can be seen that in the first set of signals, the original signals, there are some missing values. After the linear interpolation is applied, these missing values disappear. The inserted values can be seen in the second set of signals, the interpolated signals. To finish, in the third box the expanded signals are shown, where the previously interpolated signals are extended to the maximum length (from 146 to 212 frames in this case).

The Common Spatial Patterns algorithm is applied after the sets of signals are defined for every video. The CSP algorithm (first mentioned in [44] as Fukunaga-Koontz Transform) tries to find an optimum spatial filter to reduce the dimensionality of the original signals, which can be considered as an extension of Principal Component Analysis (PCA). It is applied in signal processing and commonly used for electroencephalography (EEG) systems in Brain Computer Interface (BCI) applications, although this time it is used for feature extraction in a SLR task. This algorithm works with just two classes, where the CSP filter maximizes the difference of the variances between the targets. The signals from both classes are projected with the CSP filter and while the variance of the filtered signals of one of the classes is maximized, the variance for the other class is minimized.

In order to perform the classification some features are extracted from the projected signals after applying the CSP algorithm. As CSP filter focuses on the variances of the signals, first these variances are taken as features. When executing the CSP algorithm the value of the  $q$  variable has to be selected, which represents how many feature vectors are considered in the projection. The feature vectors of the spatial filter are sorted by variance, and the  $q$  first and  $q$  last vectors are selected, which produce the smallest variance for one class and the largest variance for the other class, as it can be seen in the example shown in Fig 5. This way,  $2 \times q$  variance values are used as features for classification.

In the Fig 5 the vectors are shown in pairs, which are the vectors  $i$  and  $i + q$  that differentiate the variances the most. As it can be seen, while for  $a1$  and  $a4$  the largest variances belong to *Opaque* class, for their pairs,  $b1$  and  $b4$ , the largest variances belong to *Red* class.

In addition to these variance values, other features are extracted from the projected signals: the maximum value, the minimum value and the interquartile range (IQR). These values extracted from the signals are used, along with the previously mentioned variances, as features in the classification process.



**Fig 5. Boxplot of variances of different projection vectors, by class.**

<https://doi.org/10.1371/journal.pone.0276941.g005>

**3.2.3 Classification.** For the classification phase different classifiers have been used: bagging (BAGG, BAGG25), decision trees (J48, Random Forest (RF)), K-Nearest Neighbors (1NN, 3NN, 5NN), Naive Bayes (NB), Support Vector Machine (SVM) and a Multilayer Perceptron (MLP). The details of the parameters of the used classifiers are displayed in [Table 3](#). A comparison between the results obtained with these classifiers is made and the best performers are selected.

[Table 4](#) shows the different values that the parameters used throughout the pipeline can take. In total, 80 different configurations have been used to perform the tests, combining the values of the parameters.

As the CSP method only accepts two classes as input, all the tests have been carried out pairwise (861 tests have been performed for each configuration,  $42 \times 41 \div 2$ ). Given that the gestures in the dataset are performed by 10 different signers, it has been decided to perform a leave-one-person-out cross validation saving one person for testing each time, and using the



**Table 3. Used classifiers and their parameters.**

Classifier	Parameters
Bagging (BAGG, BAGG25)	<ul style="list-style-type: none"> <li>• Number of iterations: 10 (BAGG), 25 (BAGG25)</li> <li>• Base classifier: REPTree</li> <li>• Size of each bag, percentage: 100</li> </ul>
J48	<ul style="list-style-type: none"> <li>• Confidence factor for pruning: 0.25</li> <li>• Number of folds: 3</li> <li>• Minimum number of instances per leaf: 2</li> <li>• Pruning: True</li> <li>• Subtree raising: True</li> </ul>
Random Forest (RF)	<ul style="list-style-type: none"> <li>• Maximum depth: unlimited</li> <li>• Number of trees: 100</li> </ul>
K-Nearest Neighbours (KNN)	<ul style="list-style-type: none"> <li>• Number of neighbours: 1 (1NN), 3 (3NN), 5 (5NN)</li> <li>• Distance weighting: No</li> <li>• Nearest neighbours search algorithm: LinearNNSearch</li> <li>• Distance: Euclidean</li> </ul>
Naive Bayes (NB)	<ul style="list-style-type: none"> <li>• Use kernel estimator: False</li> <li>• Use supervised discretization: False</li> </ul>
Support Vector Machine (SVM)	<ul style="list-style-type: none"> <li>• C regularization parameter: 1</li> <li>• Kernel: radial basis function (RBF)</li> <li>• Tolerance for stopping: 0.001</li> </ul>
MultiLayer Perceptron (MLP)	<ul style="list-style-type: none"> <li>• Learning rate: 0.3</li> <li>• Momentum: 0.2</li> <li>• Hidden layers: (attributes + classes)/2</li> <li>• Training epochs: 500</li> <li>• Validation threshold: 20</li> </ul>

<https://doi.org/10.1371/journal.pone.0276941.t003>

rest for training, calculating the accuracy value of the model with the mean value of every test set. This way, it is ensured that the model is not overfitting to the people it is trained with.

## 4 Experimental results

The obtained results are presented in [Table 5](#). The values shown are achieved calculating the mean value for each configuration, which are obtained taking into account every pairwise test that has been performed.

The results show that the best mean results are obtained with  $q = 15$ , 5-NN classifier, and  $(x, y, z)$  coordinates as features in both situations, with RGB color space and black and white images. Although 5-NN obtains better results, 1-NN and 3-NN achieve high accuracy values too, being K-Nearest Neighbors classifier the one which gets better outcomes. Regarding the rest of the classifiers, J48 obtains the lowest accuracy values, followed by Naive Bayes and

**Table 4. Configuration of the classification.**

Color space	original—black & white
Classifiers	BAGG—BN—J48—KNN—NB—RF—SVM—MLP
q value	10—15
Used information	variance, max, min, IQR
Used coordinates	$(x,y)$ — $(x,y,z)$

<https://doi.org/10.1371/journal.pone.0276941.t004>

Table 5. Obtained results with different configurations.

			BAGG	BAGG25	BN	BN2	J48	1-NN	3-NN	5-NN	NB	RF	SVM	MLP
RGB	(x,y,z)	q = 10	0.9004	0.9031	0.8981	0.8982	0.8040	0.9023	0.9020	0.9013	0.9025	0.8842	0.9100	0.8994
		q = 15	0.9263	0.9297	0.9293	0.9292	0.8109	0.9485	0.9497	<b>0.9502</b>	0.8981	0.9045	0.9454	0.9473
	(x,y)	q = 10	0.9186	0.9224	0.9188	0.9213	0.8058	0.9284	0.9288	0.9278	0.9282	0.8958	0.9357	0.9232
		q = 15	0.9403	0.9417	0.9338	0.9350	0.8056	0.9480	0.9489	0.9490	0.9241	0.9229	0.9447	0.9456
Black/white	(x,y,z)	q = 10	0.9524	0.9531	0.9463	0.9465	0.8327	0.9522	0.9520	0.9509	0.9429	0.9383	0.9554	0.9506
		q = 15	0.9731	0.9756	0.9754	0.9761	0.8434	0.9829	0.9837	<b>0.9843</b>	0.9238	0.9555	0.9813	0.9808
	(x,y)	q = 10	0.9686	0.9711	0.9642	0.9659	0.8263	0.9723	0.9731	0.9733	0.9562	0.9489	0.9752	0.9695
		q = 15	0.9786	0.9804	0.9754	0.9753	0.8210	0.9816	0.9826	0.9832	0.9454	0.9659	0.9791	0.9800

<https://doi.org/10.1371/journal.pone.0276941.t005>

Random Forest. In order to analyse the information and draw conclusions, in Table 6 some statistics are shown which resume the results of Table 5 for each parameter value.

According to the obtained results, it can be concluded that MediaPipe works better on the black and white videos than on the original RGB videos. As the signers wear colorful gloves, it has been noticed that MediaPipe is not very accurate sometimes. For the purpose of trying to improve its performance, the original videos have been converted to black and white and as the results show the goal have been achieved as the accuracy values have become better. When it comes to the coordinates used as features, similar accuracy values are obtained with both options. Although using just (x, y) coordinates better mean accuracy value is achieved as it is shown in Table 6, it has already been mentioned that the best accuracy values have been obtained with (x, y, z) coordinates, which are highlighted in Table 5. Thus, not meaningful difference is perceived with respect to the coordinates chosen for the classification. However, since fewer features are used when only taking into account (x, y) coordinates, it can be said that this approach is preferable. Regarding the selected value for q parameter when applying the CSP algorithm, which determines how many feature vectors are used in the projection, better outcomes are attained with q = 15.

In short, the best mean accuracy values are obtained with these parameter values for each color space, as highlighted in Table 5.

$$\text{RGB} \left\{ \begin{array}{l} 5NN \\ (x, y, z) \\ q = 15 \end{array} \right. \quad \text{Black/white} \left\{ \begin{array}{l} 5NN \\ (x, y, z) \\ q = 15 \end{array} \right.$$

These accuracy values are not enough to compare the differences between the tested classes. As a way to analyze the results obtained for each of the classes in the database, Table 7 shows the mean values obtained for each class, which have been calculated with the accuracy values

Table 6. Obtained results for each parameter value.

	Color space		Used coordinates		q variable for CSP	
	RGB	B/W	(x,y,z)	(x,y)	q = 10	q = 15
Mean	0.9139	0.9546	0.9288	0.9397	0.9250	0.9436
Median	0.9237	0.9691	0.9442	0.9468	0.9286	0.9487
Stdev	0.0374	0.0405	0.0431	0.0442	0.0418	0.0442

<https://doi.org/10.1371/journal.pone.0276941.t006>

Table 7. Mean accuracy values obtained with the best configuration (RGB and B/W color spaces) for each class.

	Opaque	Red	Green	Yellow	Bright	Light-blue	Colors
RGB	0.9614	0.9554	0.9473	0.9720	0.9645	0.9203	0.9257
B/W	0.9927	0.9862	0.9787	0.9880	0.9959	0.9605	0.9575
	Red 2	Women	Enemy	Son	Man	Away	Drawer
RGB	0.9488	0.9457	0.9182	0.9055	0.8967	0.9596	0.9401
B/W	0.9829	0.9881	0.9849	0.9847	0.9795	0.9865	0.9890
	Born	Learn	Call	Skimmer	Bitter	Sweet-milk	Milk
RGB	0.8463	0.9564	0.9565	0.9584	0.9282	0.9470	0.9571
B/W	0.9739	0.9839	0.9856	0.9963	0.9862	0.9834	0.9882
	Water	Food	Argentina	Uruguay	Country	Last name	Where
RGB	0.9576	0.9407	0.9755	0.9879	0.9724	0.9576	0.9722
B/W	0.9839	0.9776	0.9846	0.9978	0.9846	0.9781	0.9876
	Birthday	Hungry	Ship	None	Name	Patience	Perfume
RGB	0.9726	0.9477	0.9651	0.9653	0.9858	0.9606	0.9483
B/W	0.9907	0.9838	0.9893	0.9864	0.9822	0.9902	0.9774
	Deaf	Candy	Chewing-gum	Shut down	Buy	Realize	Find
RGB	0.8966	0.9809	0.9805	0.9713	0.9451	0.9483	0.9664
B/W	0.9876	0.9922	0.9917	0.9878	0.9644	0.9775	0.9915

<https://doi.org/10.1371/journal.pone.0276941.t007>

Table 8. Statistics of results obtained with best parameter settings.

	MAX	MIN	Q1	MEDIAN	Q3
RGB	0.9879	0.8463	0.9453	0.9568	0.9661
Black & White	0.9978	0.9575	0.9824	0.9859	0.9888

<https://doi.org/10.1371/journal.pone.0276941.t008>

of all the test pairs in which each class has participated. The displayed values are achieved with the parameter values mentioned above, which produce the best setting.

At first glance, there is a definite distinction between using the original RGB videos and those that have been converted to black and white. For black and white videos, classes like *URUGUAY*, *SKIMMER* or *BRIGHT* get a high accuracy value,  $> 0.995$ . Other classes, such as *COLORS*, *LIGHT-BLUE* or *BUY*, on the other hand, remain for  $0.95 \sim 0.96$  values. In the case of RGB videos, the best classified classes are *URUGUAY*, which coincides in both color spaces, and *NAME*, while the worst classified are *DEAF*, *MAN* and *BORN*, which drops to a value of 0.84.

In Table 8 several statistics are shown to summarize the results of Table 7. As mentioned before, black and white videos are better to perform the classification, which is evident from these statistics. The accuracy values of all the classes are between 0.8463 – 0.9879 for RGB videos and 0.9575 – 0.9978 for black and white videos. The first quartile value shows that most of the classes get higher than 0.9453 and 0.9824 accuracy values for RGB and black and white videos respectively. Therefore, it can be concluded that there is not a remarkable difference between the tested classes.

## 5 Conclusion

In this paper, a Sign Language Recognition approach is presented, where videos of an Argentinian Sign Language dataset are used. For each video frame several hand landmarks are

obtained applying MediaPipe technology. A set of signals is created from each video using these hand landmarks. The CSP algorithm is used to transform these signals and, after extracting some features from them (variance, maximum, minimum and IQR values), classification is carried out. Different classifiers have been employed for classification. It must be mentioned that the presented approach is non-intrusive; signers do not need to have any sort of gadget attached to them, which makes the system more comfortable for them. The obtained results are between 0.90 and 0.95, yielding higher accuracy values after converting the original videos to black and white color space. The classification results are therefore promising.

While deep learning approaches are currently state-of-the-art in practically all fields of research, their hyperparameters still need to be fine tuned, which requires running many training epochs with each set of candidate hyperparameter values. One benefit of our approach is that the CSP has a closed form and therefore it is possible to compute it without using iterative methods. There are fewer hyperparameters in the research herein presented—just five—than in a typical deep learning hyperparameter tuning task (see [Table 4](#)).

Although the dataset used is rather limited, with a small number of signs, it is proven that the use of CSP can be beneficial for classification tasks. However, there is still a lot of work to be done in the field of sign language recognition, as being able to recognize a limited number of signs is far away from obtaining a system capable of operating as an interpreter. Therefore, further research should be carried out in this area and, more specifically, in the aforementioned field of sign translation.

## 5.1 Future work

Several tasks have been identified as future work. Some of these ideas are presented below.

- In the LSA64 dataset the signers wear colorful gloves to make the hand segmentation task easier. As specified, the presented approach is non-intrusive, thus these gloves are not required. Instead of helping, the gloves could be more of a hindrance than an aid when applying MediaPipe. In order to avoid this issue, another database should be used, one in which the signers are not wearing gloves and their hands are clearly visible. We are currently actively working in creating a small database of bare hand configurations and gestures for the Spanish Sign Language.
- Adding facial information is important too. Experts in sign languages emphasize the importance of this channel of information when communicating. MediaPipe includes the capability of obtaining face landmarks from videos with its FaceMesh solution. However, as previously mentioned, the participants do not use the proper face expressions when performing the signs, in the videos used in this work they focus on the movements of the hands. Another database should be selected, where signers actually change their facial expression depending on the sign, to add this information into the classification pipeline.
- The used database includes videos of both one and two-handed signs. In the presented approach only the one-handed signs are used, excluding those signs that make use of both hands to perform them. Two-handed signs should also be added, making the classification more challenging.
- In an effort to improve the performance of MediaPipe, original videos have been converted to black and white color space. Other preprocessing approaches could be applied, in order to establish the optimum image configuration for MediaPipe and thus, obtain more accurate hand landmarks positions.

To sum up, it has been shown that the Common Spatial Patterns algorithm, which is typically used in processing of physiological signals, can be successfully applied in other domains, i. e. Sign Language Recognition, as a feature extraction method combined with technologies like MediaPipe.

It is also noteworthy that, instead of working over the CSP features, it would also be possible to work over the CSP transformed signals and apply other techniques. For example, deep learning could be applied to these transformed signals that have been projected into a lower dimensional space.

## Author Contributions

**Conceptualization:** Basilio Sierra.

**Investigation:** Itsaso Rodríguez-Moreno.

**Software:** Itsaso Rodríguez-Moreno.

**Supervision:** José María Martínez-Otzeta, Basilio Sierra.

**Validation:** José María Martínez-Otzeta, Izaro Goienetxea.

**Writing – original draft:** Itsaso Rodríguez-Moreno.

**Writing – review & editing:** José María Martínez-Otzeta, Izaro Goienetxea, Basilio Sierra.

## References

1. Perlman M, Little H, Thompson B, Thompson RL. Iconicity in signed and spoken vocabulary: a comparison between American Sign Language, British Sign Language, English, and Spanish. *Frontiers in psychology*. 2018 Aug 14; 9:1433. <https://doi.org/10.3389/fpsyg.2018.01433> PMID: 30154747
2. Klima ES, Bellugi U. *The signs of language*. Harvard University Press; 1979.
3. Padden CA, Gunsauls DC How the alphabet came to be used in a sign language. *Sign Language Studies*. 2003 Oct 1:10–33. <https://doi.org/10.1353/sls.2003.0026>
4. Lugaresi C, Tang J, Nash H, McClanahan C, Uboweja E, Hays M, et al. MediaPipe: A framework for building perception pipelines. *arXiv preprint arXiv:190608172*. 2019.
5. Fukunaga K, Koontz WL. Application of the Karhunen-Loève Expansion to Feature Selection and Ordering. *IEEE Transactions on computers*. 1970; 100(4):311–318. <https://doi.org/10.1109/T-C.1970.222918>
6. Alotaiby TN, Alshebeili SA, Aljafar LM, Alsabhan WM. ECG-based subject identification using common spatial pattern and SVM. *Journal of Sensors*. 2019; 2019. <https://doi.org/10.1155/2019/8934905>
7. Kim P, Kim KS, Kim S. Using common spatial pattern algorithm for unsupervised real-time estimation of fingertip forces from sEMG signals. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2015. p. 5039–5045.
8. Li X, Fang P, Tian L, Li G. Increasing the robustness against force variation in EMG motion classification by common spatial patterns. In: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE; 2017. p. 406–409.
9. Shapiro J, Savransky D, Ruffio JB, Ranganathan N, Macintosh B. Detecting Planets from Direct-imaging Observations Using Common Spatial Pattern Filtering. *The Astronomical Journal*. 2019; 158(3):125. <https://doi.org/10.3847/1538-3881/ab3642>
10. Rodríguez-Moreno I, Martínez-Otzeta JM, Goienetxea I, Rodríguez IR, Sierra B. Shedding Light on People Action Recognition in Social Robotics by Means of Common Spatial Patterns. *Sensors*. 2020; 20(8):2436. <https://doi.org/10.3390/s20082436> PMID: 32344755
11. Rodríguez-Moreno I, Martínez-Otzeta JM, Goienetxea I, Sierra B. Sign Language Recognition by Means of Common Spatial Patterns. In: 2021 The 5th International Conference on Machine Learning and Soft Computing (ICMLSC'21). In press; 2021.
12. Koller O. Quantitative survey of the state of the art in sign language recognition. *arXiv preprint arXiv:200809918*. 2020.
13. Rastgoo R, Kiani K, Escalera S. Sign language recognition: A deep survey. *Expert Systems with Applications*. 2020; p. 113794.

14. Cheok MJ, Omar Z, Jaward MH. A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*. 2019; 10(1):131–153. <https://doi.org/10.1007/s13042-017-0705-5>
15. Von Agris U, Knorr M, Kraiss KF. The significance of facial features for automatic sign language recognition. In: 2008 8th IEEE International Conference on Automatic Face & Gesture Recognition. IEEE; 2008. p. 1–6.
16. Zhou H, Zhou W, Zhou Y, Li H. Spatial-temporal multi-cue network for continuous sign language recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. p. 13009–13016.
17. Kaluri R, CH PR. Optimized feature extraction for precise sign gesture recognition using self-improved genetic algorithm. *International Journal of Engineering and Technology Innovation*. 2018; 8(1):25–37.
18. Rosero-Montalvo PD, Godoy-Trujillo P, Flores-Bosmediano E, Carrascal-García J, Otero-Potosi S, Benitez-Pereira H, et al. Sign language recognition based on intelligent glove using machine learning techniques. In: 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM). IEEE; 2018. p. 1–5.
19. Kakoty NM, Sharma MD. Recognition of sign language alphabets and numbers based on hand kinematics using a data glove. *Procedia Computer Science*. 2018; 133:55–62. <https://doi.org/10.1016/j.procs.2018.07.008>
20. Ma Y, Zhou G, Wang S, Zhao H, Jung W. Signfi: Sign language recognition using wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*. 2018; 2(1):1–21.
21. Kumar P, Gauba H, Roy PP, Dogra DP. Coupled HMM-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters*. 2017; 86:1–8. <https://doi.org/10.1016/j.patrec.2016.12.004>
22. Kumar P, Gauba H, Roy PP, Dogra DP. A multimodal framework for sensor based sign language recognition. *Neurocomputing*. 2017; 259:21–38. <https://doi.org/10.1016/j.neucom.2016.08.132>
23. Ibrahim NB, Selim MM, Zayed HH. An automatic arabic sign language recognition system (ArSLRS). *Journal of King Saud University-Computer and Information Sciences*. 2018; 30(4):470–477. <https://doi.org/10.1016/j.jksuci.2017.09.007>
24. Pu J, Zhou W, Li H. Iterative alignment network for continuous sign language recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 4165–4174.
25. Rao GA, Syamala K, Kishore P, Sastry A. Deep convolutional neural networks for sign language recognition. In: 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES). IEEE; 2018. p. 194–197.
26. Barbhuiya AA, Karsh RK, Jain R. CNN based feature extraction and classification for sign language. *Multimedia Tools and Applications*. 2021; 80(2):3051–3069. <https://doi.org/10.1007/s11042-020-09829-y>
27. Parelli M, Papadimitriou K, Potamianos G, Pavlakos G, Maragos P. Exploiting 3D hand pose estimation in deep learning-based sign language recognition from RGB videos. In: European Conference on Computer Vision. Springer; 2020. p. 249–263.
28. Cao Z, Hidalgo G, Simon T, Wei SE, Sheikh Y. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:181208008*. 2018.
29. Gadekallu TR, Alazab M, Kaluri R, Maddikunta PKR, Bhattacharya S, Lakshmana K, et al. Hand gesture classification using a novel CNN-crow search algorithm. *Complex & Intelligent Systems*. 2021; 7(4):1855–1868. <https://doi.org/10.1007/s40747-021-00324-x>
30. Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures*. 2016; 169:1–12. <https://doi.org/10.1016/j.compstruc.2016.03.001>
31. Li D, Rodriguez C, Yu X, Li, H. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision; 2020. p. 1459–1469.
32. Duarte A, Palaskar S, Ventura L, Ghadiyaram D, DeHaan K, Metze F, et al. How2Sign: A large-scale multimodal dataset for continuous american sign language. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2021. p. 2735–2744.
33. Sincar OM, Junior J, Jacques CS, Escalera S, Keles HY. Chalearn LAP large scale signer independent isolated sign language recognition challenge: Design, results and future research. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 3472–3481.
34. Varol G, Momeni L, Albanie S, Afouras T, Zisserman A. Scaling up sign spotting through sign language dictionaries. *International Journal of Computer Vision*; 2018; 130(6):1416–1439. <https://doi.org/10.1007/s11263-022-01589-6>

35. Li D, Xu C, Yu X, Zhang K, Swift B, Suominen H, et al. TSPNet: Hierarchical feature learning via temporal semantic pyramid for sign language translation. *Advances in Neural Information Processing Systems*;2020;33:12034–12045.
36. Camgoz NC, Hadfield S, Koller O, Ney H, Bowden, R. Neural sign language translation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2018. p. 7784-7793.
37. Forster J, Schmidt C, Hoyoux T, Koller O, Zelle U, Piater JH, et al. RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus. In: *LREC*. vol. 9; 2012. p. 3785–3789.
38. Chai X, Wang H, Chen X. The DEVISIGN large vocabulary of chinese sign language database and baseline evaluations. Technical report VIPL-TR-14-SLR-001 Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS. 2014.
39. Von Agris U, Kraiss KF. Towards a video corpus for signer-independent continuous sign language recognition. *Gesture in Human-Computer Interaction and Simulation*, Lisbon, Portugal, May. 2007;11.
40. Rastgoo R, Kiani K, Escalera S. Hand sign language recognition using multi-view hand skeleton. *Expert Systems with Applications*. 2020; 150:113336. <https://doi.org/10.1016/j.eswa.2020.113336>
41. Adaloglou N, Chatzis T, Papastratis I, Stergioulas A, Papadopoulos GT, Zacharopoulou V, et al. A comprehensive study on sign language recognition methods. *arXiv preprint arXiv:200712530*. 2020.
42. Ronchetti F, Quiroga F, Estrebow CA, Lanzarini LC, Rosete A. LSA64: an Argentinian sign language dataset. In: *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*.; 2016.
43. Zhang F, Bazarevsky V, Vakunov A, Tkachenka A, Sung G, Chang CL, et al. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv preprint arXiv:200610214*. 2020.
44. Fukunaga K, Koontz WL. Application of the Karhunen-Loeve expansion to feature selection and ordering. *IEEE Transactions on computers*. 1970; 100(4):311–318. <https://doi.org/10.1109/T-C.1970.222918>





# HAKA: HierArchical Knowledge Acquisition in a Sign Language Tutor

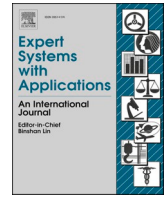
<b>Title:</b>	HAKA: HierArchical Knowledge Acquisition in a Sign Language Tutor
<b>Authors:</b>	I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra
<b>Journal:</b>	Expert Systems with Applications
<b>Publisher:</b>	Elsevier
<b>DOI:</b>	10.1016/j.eswa.2022.119365
<b>Year:</b>	2022
<b>Source of impact:</b>	WOS (JCR)
<b>Category:</b>	COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE
<b>Impact index:</b>	8.665 (Q1)
<b>Position:</b>	21/145





Contents lists available at ScienceDirect

## Expert Systems With Applications

journal homepage: [www.elsevier.com/locate/eswa](http://www.elsevier.com/locate/eswa)

## HAKA: HierArchical Knowledge Acquisition in a sign language tutor

Itsaso Rodríguez-Moreno <sup>\*,1</sup>, José María Martínez-Otzeta, Basilio Sierra

Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Manuel Lardizabal 1, Donostia-San Sebastián 20018, Gipuzkoa, Spain

## ARTICLE INFO

## Keywords:

Sign language  
Language tutor  
Action recognition  
Procrustes similarity  
Multidimensional scaling

## ABSTRACT

Communication between people from different communities can sometimes be hampered by the lack of knowledge of each other's language. A large number of people needs to learn a language in order to ensure a fluid communication or want to do it just out of intellectual curiosity. To assist language learners' needs tutor tools have been developed. In this paper we present a tutor for learning the basic 42 hand configurations of the Spanish Sign Language, as well as more than one hundred of common words. This tutor registers the user image from an off-the-shelf webcam and challenges her to perform the hand configuration she chooses to practice. The system looks for the configuration, out of the 42 in its database, closest to the configuration performed by the user, and shows it to her, to help her to improve through knowledge of her errors in real time. The similarities between configurations are computed using Procrustes analysis. A table with the most frequent mistakes is also recorded and available to the user. The user may advance to choose a word and practice the hand configurations needed for that word. Sign languages have been historically neglected and deaf people still face important challenges in their daily activities. This research is a first step in the development of a Spanish Sign Language tutor and the tool is available as open source. A multidimensional scaling analysis of the clustering of the 42 hand configurations induced by Procrustes similarity is also presented.

## 1. Introduction

People from interacting communities where different languages are spoken have the need to overcome this communication barrier. Along history these interactions have made languages disappear, evolve or become dominant in a region or cultural domain. In spite of the current trends of increasing globalization and interconnectedness, there are more than 7,000 living languages in the world (Eberhard, Simons, & Fennig, 2022).

Given these figures it is safe to conclude that the knowledge of languages not natively acquired is a must for many people in the world. In addition to those who have that need, there are also people who want to be fluent in another language just out of intellectual curiosity. The demand of individuals as well as the public and private sector made the global market size of the language industry at around 49.6 billion U.S. dollars in 2019 (Mazereanu, 2019).

Information technologies are nowadays a fundamental part of the process of teaching, studying and/or practicing a new language. Even in the traditional teacher-student setting, videoconferencing applications

allow for remote lessons weakening the constraints faced when the teacher and the student needed to share the same spatial location. Recording of the lessons also permits a time managing more suited to the needs of the learner. But the most revolutionary changes have come with the use of software engineering practices and artificial intelligence techniques. Companies who deploy websites with user progress customization, tailored tutoring and automatic voice recognition are competing with companies with a more classic approach based on onsite teaching by native speakers.

In this paper we report on the ongoing work in a Spanish Sign Language tutor which could capture the user image in a standard webcam and recognize the hand configuration or sign she is performing and give feedback about her progress. Sign languages are used by the hearing-impaired community, mainly for communication among their members, given the lack of knowledge outside their community. A sign is composed by a succession of hand configurations, along with body movements or facial expressions. Our approach is bottom-up, building the hand configuration recognizer before and as a part of the sign recognizer. In this way, compared with end-to-end deep learning

\* Corresponding author.

E-mail addresses: [itsaso.rodriguez@ehu.eus](mailto:itsaso.rodriguez@ehu.eus) (I. Rodríguez-Moreno), [josemaria.martinez@ehu.eus](mailto:josemaria.martinez@ehu.eus) (J.M. Martínez-Otzeta), [b.sierra@ehu.eus](mailto:b.sierra@ehu.eus) (B. Sierra).

<sup>1</sup> ORCID: 0000-0001-8471-9765.

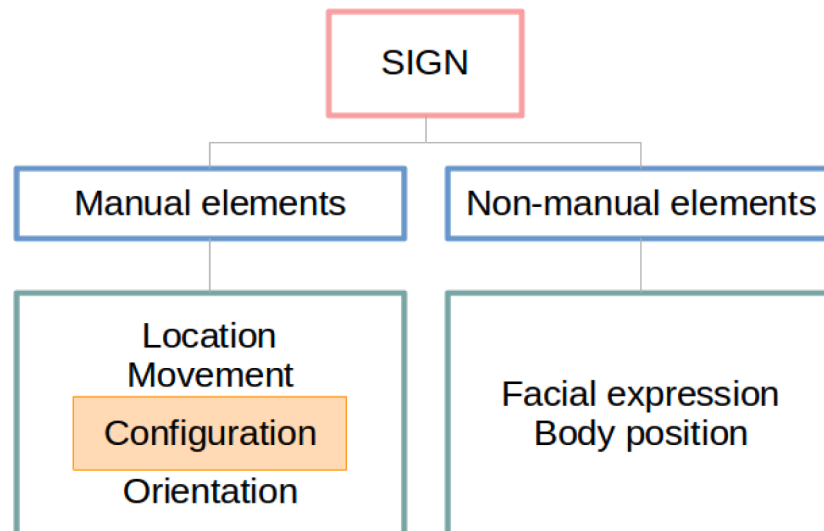


Fig. 1. Elements which compose a sign.

approaches, it is possible to have a better interpretability of the tutor decisions and the learner can be informed of why she has made a mistake and not only that she has made a mistake. The hand configuration recognizer is based in the Procrustes distance between the user configuration and the stored models, and therefore it is very easy to add or remove another model to the stored library.

The paper is organised as follows: the next section is devoted to other research related to the current work. Then the proposed approach is presented and the following section presents the functionalities of the developed application. The paper closes with a discussion about the insights gained from Procrustes analysis and outlines further work.

## 2. Related work

Computer-assisted language learning is an active interdisciplinary field of research which encompasses many topics covering from pedagogy to artificial intelligence (Chen, Zou, & Su, 2021). In (Zhang & Zou, 2022) the authors find that five main types of technology are employed for language learning: speech-to-text and text-to-speech recognition, mobile learning, socialized learning, multimedia learning and game-based learning. They also show that these technologies make it easier the delivery of content and facilitate interactions, while at the same time promote language practising and restructure teaching approaches.

While the majority of languages could benefit from speech-to-text and text-to-speech technologies, sign languages processing needs to focus on gestures instead of speech. Signs are performed mainly with the hands, although general body movement and face expressions may also convey meaning. In order to recognize the sign, some data capture system is needed. The use of electronic gloves that detect or record the hand position (De Marco & Foulds, 2003; Ahmed, Zaidan, Zaidan, Salih, & Lakulu, 2018) or of colored gloves that can be segmented from the whole image (Wang & Popović, 2009; Azar & Seyedarabi, 2020) is being made redundant thanks to the latest advances in machine learning, which permit hand pose estimation without extra equipment. In (Quinn & Olszewska, 2019) the authors present a system designed to detect and recognize British Visual Language signs, using Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005) and multi-class SVMs (Cortes & Vapnik, 1995). Hidden Markov Models (HMMs) (Baum & Petrie, 1966) have also been applied in this field of research. In (Azar & Seyedarabi, 2020) a dynamic Persian sign language recognition system is presented, where Gaussian HMMs are used as modeling tools for hand trajectories of signers. The hand segmentation is facilitated by the users wearing white gloves. Another approach based on skin segmentation is described

in (Roy, Kumar, & Kim, 2021), where the Camshift algorithm (Bradski, 1998) is employed to track the hand and the different trajectories when performing ASL signs are classified using HMMs.

Deep learning techniques are widely used nowadays due to their effectiveness in all kind of domains and sign language processing is no exception. Researchers in different sign languages have made use of deep learning to achieve significant progress in sign recognition. A real-time system for recognition of American Sign Language using a convolutional neural network is presented in (Taskiran, Killioglu, & Kahraman, 2018), while in (Boháček & Hrdz, 2022) the authors perform a word-level sign recognition for ASL using pose-based transformers (Vaswani, et al., 2017). In (Sevli & Kemaloglu, 2020) the authors also use convolutional neural networks for the recognition of digits in Turkish sign language, while in (Aktaş, Gökberk, & Akarun, 2019) a ResNet architecture (He, Zhang, Ren, & Sun, 2016) is employed to recognize Turkish non-manual signs, as facial expressions and head movements. Other sign languages for which deep learning approaches have been applied include Indian (Sharma, Sharma, Saxena, Singh, & Sadhya, 2021), Chinese (Gao, et al., 2021), Indonesian (Fadlilah et al., 2021) and Brazilian (Rocha, Lensk, Ferreira, & Ferreira, 2020), to name just a few. Comprehensive surveys of the current deep-learning-based research can be found in (Rastgoo, Kiani, & Escalera, 2021) and (Al-Qurishi, Khalid, & Souissi, 2021).

While the performance of deep learning systems is impressive, they often lack the ability to explain their conclusions in human-understandable terms (Fazi, 2021). They are also often presented as universal learners from low-processed data, with less need for feature engineering, although adding experts' insight into a traditional approach could be competitive against them (Jiang, et al., 2018).

Furthermore, applications using some definition of distance between static or dynamic gestures have been proven successful in some domains. In (Ibañez, Soria, Teyseyre, Rodríguez, & Campo, 2017) the authors propose a lightweight approach to gesture recognition by encoding the movements of 3D joints coordinates provided by a Kinect camera into a string. Then they use techniques of string matching to tackle the problem. Another classical statistical technique, Procrustes analysis (Gower, 1975; Dryden & Mardia, 2016), has been employed to detect hand-over-face expressions (Révy, Hadházi, & Hullám, 2022).

The hand gesture recognition module of the Spanish Sign Language tutor system that we are developing builds on the definition of distance between the performed hand configuration and the elements of a database. One of the advantages of this approach is that the system could show the user which is the gesture, from the database, most similar to

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42

Fig. 2. The 42 configurations of the Spanish Sign Language. Blue: front of the side (palm) in the background. Grey: back of the side (knuckles) in the background.

the one she is performing. This might be more useful than just an error message telling the user that she is wrong but not why.

### 3. Proposed approach

A sign can be decomposed into the elements shown in Fig. 1.

- Manual elements.
  - o *Location*: the location where signs are performed, including the part of the body, the plane and the contact point. The *plane*

indicates where the sign is performed according to the distance to the body, whereas the *contact point*, if any, refers to the part of the dominant hand that touches another part of the body.

- o *Configuration*: the shape of the hand when performing a sign. In the Spanish Sign Language (*Lengua de Signos Española*, LSE) there are three different types of configurations: the configurations representing the Spanish alphabet performed by the dominant hand, the hand configurations to sign the natural numbers, and the configurations representing the phonemes, similar to the phonological system formed by the distinctive sounds of an oral language.

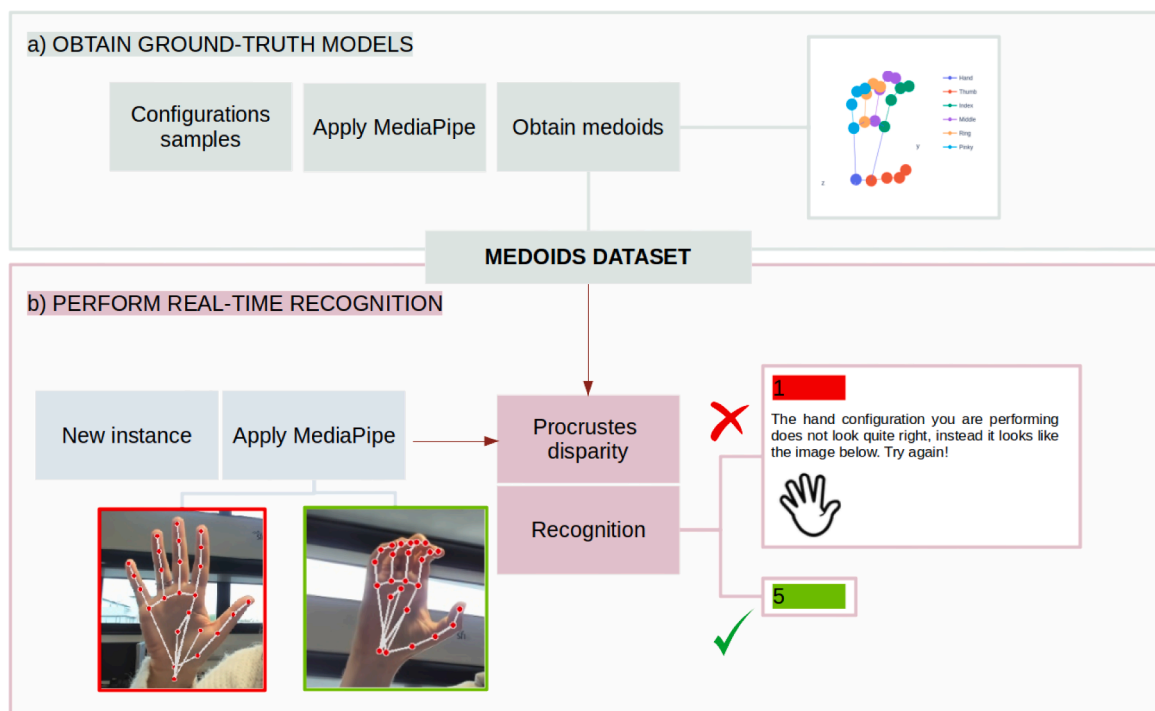


Fig. 3. Followed pipeline.

- o *Orientation*: orientation of the hands involved on the articulation of the sign with respect to the body of the signer, that is, the distinctive relative degree of rotation of the hands when signing.
- o *Movement*: the movement usually involved when performing a sign, the distinctive hand actions that form words.
- Non-manual elements. In addition to the manual elements, there are other non-manual components which are also crucial when defining a sign: the facial expression and the body position.

Since our aim is to help people to start learning the Spanish Sing Language, as a first approach we propose a basic tutor to first learn the different phonological configurations of the Spanish Sign Language, and therefore develop the skills needed to perform the signs of the language. As shown in Fig. 2, 42 different configurations are the basis for the Spanish Sign Language. Their recognition with high accuracy can be a challenging task due to the high degree of visual similarity among some subsets of configurations.

In other approaches it is usual to obtain a set of training examples and process it to create a model, knowing very little about the structure of the possible classes before the start of the learning process. In contrast, the depictions of all the possible hand configurations of the Spanish sign language are known in advance. Therefore, it is possible to create ground-truth models with the hand configurations and then compute the distance from a new input to those models, returning the most similar one. This approach is similar to KNN (more precisely 1-NN), with Procrustes disparity as distance. The process of creating the ground truth could be considered similar to prototype selection in KNN (Garcia, Derrac, Cano, & Herrera, 2012).

The followed pipeline can be seen in Fig. 3, where first we extract the hand-landmarks of the user using MediaPipe (Zhang, et al., 2020) and compare the captured shape to saved pre-defined models using Procrustes analysis in order to predict the label corresponding to the most similar model. It is worth mentioning that the process depicted in the upper part of the diagram, namely the computation of the medoids, is only executed once, as a necessary step before the tutor can be used by the public.

MediaPipe Hand Tracking solution, part of MediaPipe, is able to

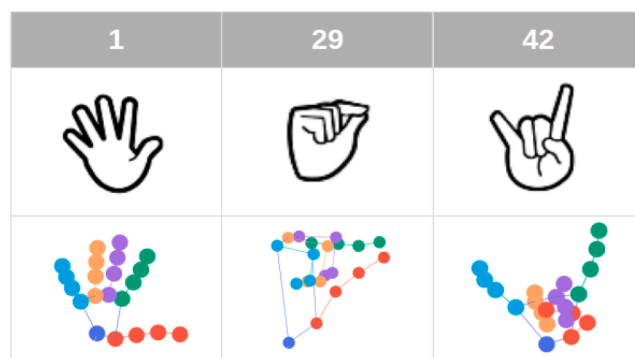


Fig. 4. Configurations and their corresponding saved medoids models.

estimate the spatial location of 21 landmarks for each hand. Each hand landmark is composed of three values, the coordinates (x,y,z), where z represents the depth with respect to the wrist. These data will be used to represent the ground truth configuration models as well as the hand configurations performed by the users.

The steps of the pipeline are explained in the following subsections.

### 3.1. Ground-truth models

To build the ground-truth models a person with basic formal education in Spanish Sign Language has performed 50 times each configuration in front of a Logitech BRIO 4 K Ultra HD Webcam and in good lighting conditions. The medoid of each set of 50 repetitions of configurations with respect to the Procrustes disparity has been computed and saved as ground-truth. In Fig. 4 some examples of the saved models are shown, along with the configuration they belong to. These are the models which are used as ground-truth to compare with the configuration the user is performing. As it can be seen, the saved pre-defined models are very similar to the actual configurations.

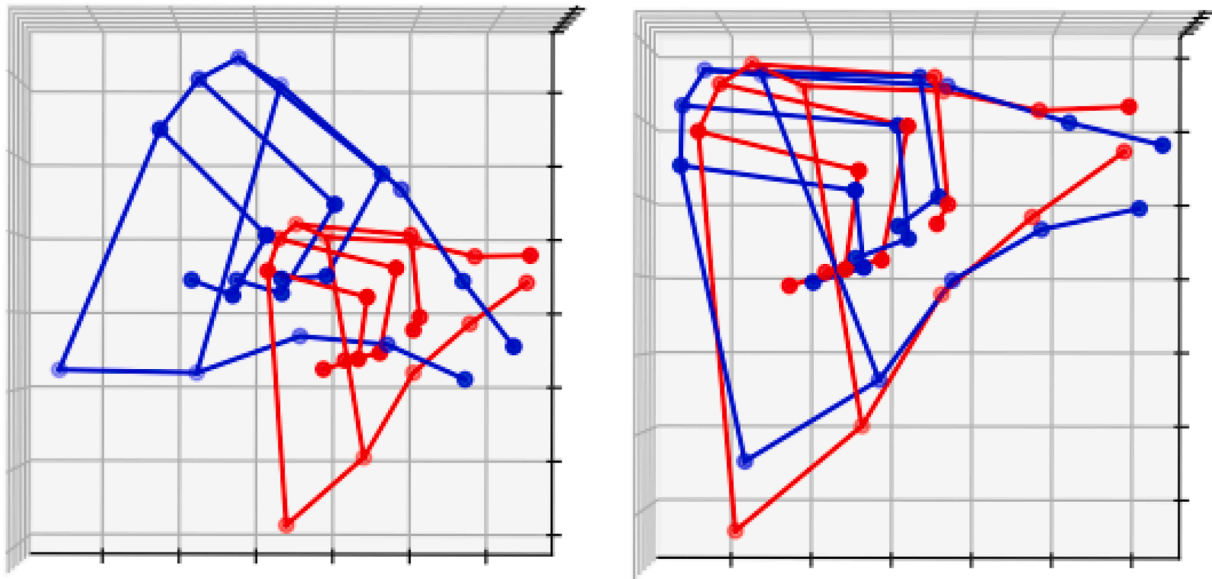


Fig. 5. Example of the transformations applied in Procrustes disparity. On the left, two original hand shapes as obtained by MediaPipe. On the right, the same hand shapes after applying the transformations to minimize their differences.

### 3.2. Procrustes analysis

The similarity between the configurations performed by the user and the saved models is calculated according to the Procrustes analysis, a statistical technique applied in areas ranging from microbiology (Tremblay, et al., 2015) to social robotics (Zabala, Rodriguez, Martínez-Otzeta, Irigoien, & Lazkano, 2021). Having two different shapes, the Procrustes analysis consists in performing a combination of transformations, including scaling, rotation and reflections, in order to minimize the difference between both shapes. After applying the transformations, the difference is calculated as indicated in Equation (1).

$$diff = \sqrt{(data_1 - data_2)^2} \tag{1}$$

In Fig. 5 an example of the hand shapes before and after the transformations can be seen. While on the left, the shapes are shown exactly as obtained after applying MediaPipe to the recording, on the right, the shapes have been transformed trying to minimize their difference. In particular, after centering both shapes around the origin, the optimal transformations are applied to the second shape (the blue shape in this case) to try to achieve as much similarity as possible with the other shape.

This way, the Procrustes similarities between the configuration that is being performed and the 42 saved models (one per configuration of

## Choose which hand you are using

Right hand  Left hand

## Choose a configuration to practice

1

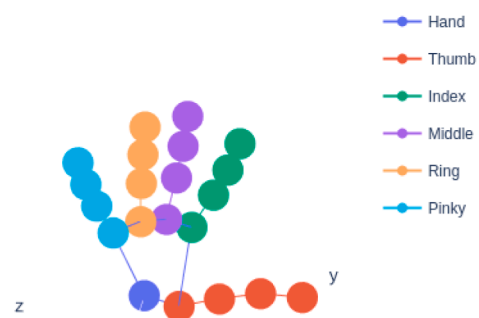




Fig. 6. A screenshot of the web application where, after choosing the configuration to perform and the hand which is being used, the image and medoid (model) for that configuration are shown.

# PERFORMANCE.

The 57.69% of the trials were well performed.

These are the ten most confusing configurations in your attempts.

	Chosen configuration	Performed configuration	% Wrong performances
	1	3	23.08
	1	2	19.23

You have selected to practice configuration 1

But you have performed configuration 2



[Continue learning](#)

Fig. 7. Performance of the user, showing the mistakes he/she has made.

the database) are calculated in order to perform the recognition. The configuration with which the lowest difference is achieved is predicted, that is, the closest configuration is used to decide the label of the performed configuration.

## 4. Functionalities

The purpose of the approach is to develop a Spanish Sign Language tutor to help people who want to learn this language. With that in mind, we have built a web application where users are able to learn how to perform the different configurations and practice those that make up different signs. The web application is available as open-source on GitHub [https://github.com/rsait/LSE\\_tutor](https://github.com/rsait/LSE_tutor).

### 4.1. Learn configurations

In the *learn configurations* functionality, the user has the option to

decide which configurations to practice and with which hand he/she is going to carry them out

Once these details are set, the selected configuration is shown to the user in two different ways:

- An image of the configuration to practice.
- A 3D plot with the hand landmarks obtained by MediaPipe corresponding to the saved model for the selected configuration.

This is expected to be helpful for the user as the visual information is displayed while he/she is performing the configurations. In Fig. 6 a screenshot of the application is shown, where the mentioned elements are shown.

After deciding the configuration and the hand to use, the user can start practicing the selected configuration. If the performance is correct, i.e. the selected configuration corresponds to the configuration the user performs, a green background is displayed to make the user aware that

## Choose which hand you are using

Right hand  Left hand

Topic:

Adjetivos (Adjectives)

Word:

Pelirrojo (Red-haired)

These are the 3 configurations you must perform:



Fig. 8. A screenshot of the web application where, after selecting a sign, the configurations that compose it are displayed.



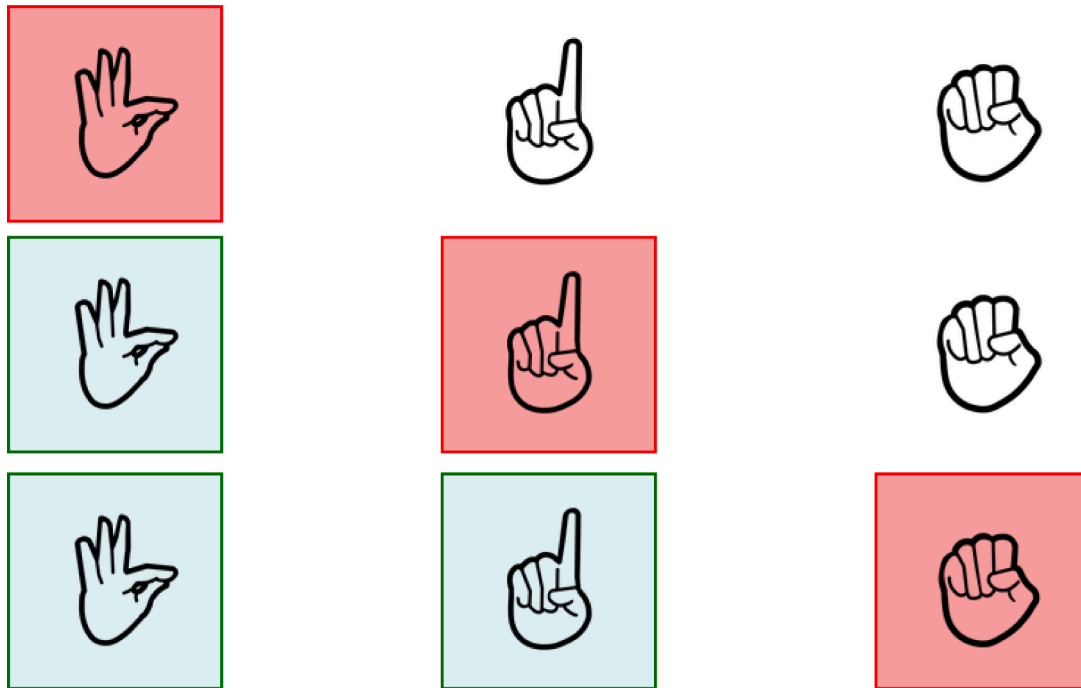


Fig. 9. Sequence of configurations that make up a sign. The red background indicates the configuration to be carried out. Once it has been correctly performed, the green background is set.

he/she is doing a good job. However, if the prediction does not match the selected configuration, apart from setting a red background, the name and the image of the configuration that the user is performing are shown. This way, the user can analyse which configuration he/she is performing and which one he/she has to perform, making the differences noticeable.

In addition, a record of the mistakes is saved (see Fig. 7). The user is able to see which have been the configurations he/she has performed worst and try to improve them. The mistakes are shown in a table, where the ten most frequent errors are listed. Each record is composed of the configuration that was selected to practice, the configuration that has been performed instead and the percentage of the wrong performances of that confusion. If a record is selected, the images of the corresponding configurations are displayed. The total percentage of the trials that were correctly performed is also indicated.

#### 4.2. Combine signs and configurations

Trying to motivate the learning of the configurations, another functionality has been added to the web application. The opportunity to practice the configurations corresponding to different signs is given. The user can select a sign among 196 different signs which have been added, and the configurations corresponding to the selected sign are shown in the same order in which they are carried out in the sign.

In Fig. 8 a screenshot of the application is displayed, showing the functionality just described. As it can be seen, there are two different dropdowns to select the sign, because these are separated into subjects. The user can choose a topic (first dropdown) and then a sign corresponding to that topic (second dropdown).

As it can be perceived in Fig. 8, when showing the configurations that have to be performed, a red background is set on the first one. The idea is to perform the configurations correctly in the same order as they are performed when the signs are actually executed. Therefore, the sequence of configurations must be performed correctly. The red background is placed on the configuration to be carried out. Once this is correctly performed, the green background is set and the red background is switched to the next one. An example sequence is shown in

Fig. 9.

#### 5. Discussion and further work

To finish, an analysis of the clustering of the 42 hand configurations induced by multidimensional scaling using Procrustes similarity as distance is also presented. In Fig. 10 the plot corresponding to the multidimensional scaling in 2D space of the configurations space is shown. This technique is useful to visualize neighbourhood relationships between objects when only distances are given. Close instances according to the distance appear also close in the graph while less similar instances are farther apart.

In this case, the multidimensional scaling is used to evaluate if the Procrustes disparity agrees with the human perception of similarity between hand configurations. In other words, if a person finds that the cluster distribution of hand configurations looks natural. This seems to be the case after looking at Fig. 10 and the configurations of Fig. 2. For example, configurations 12, 13, 14 and 15 appear very close in the plot. Looking at the shape of these configurations, they really look similar to the human eye, as they all involve the thumb finger, the index finger and the distance between them, maintaining the rest of the fingers stretched. There are other similarities that can be clearly observed in the plot:

- 16, 17, 18 and 19. This group of configurations share the position of the middle finger, the thumb and the distance between them.
- 11 and 39. In these two configurations the fist is closed and the difference is minimal, consisting of the middle finger and how it makes contact with the thumb.
- 33, 34, 35, 36, 37 and 38. These configurations share that the index and the middle finger are stretched, while the ring and the pink fingers are flexed.

Therefore, it can be concluded that the Procrustes disparity is an adequate similarity measure to compare the shape of the hands as the obtained results show that the configurations that are most confusing are considered similar. That is, the system considers similar the same as a user would consider similar.

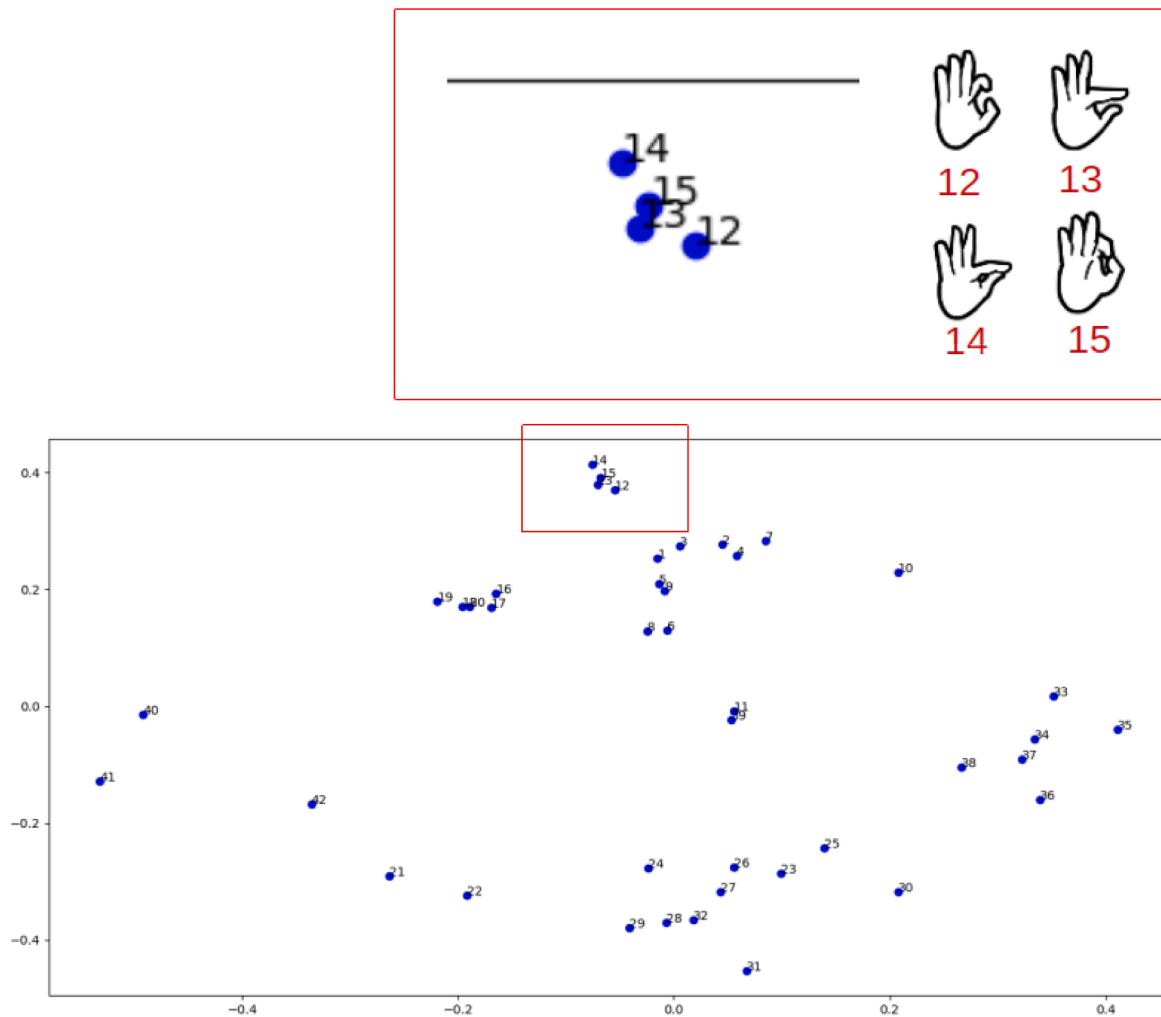


Fig. 10. Multidimensional scaling — example of similar configurations.

The addition of a new hand configuration is straightforward. The only needed modification is the addition of a new model in the model database, and the Procrustes disparity will be computed over the updated database. In this sense this approach shares similarities with K-NN or other so-called “lazy” methods, where updating knowledge does not imply retraining. However, the addition of a new model may require some work, as it must be a faithful representation of the hand configuration to be recognized. We address this issue by computing the medoid of a video sequence of a person performing 50 repetitions of each configuration. This need could represent a limitation in some contexts. Another limitation of the presented approach is that the running time is sensitive to the number of classes (the different hand configurations), since more classes will take longer to perform the Procrustes analysis. This limitation is typical in KNN-related methods and could be an issue with a big number of hand configurations. However, in the particular case of Spanish Sign Language, the number of possible configurations is 42, which can be computed in real time.

Performing the Procrustes analysis implies computing a Singular Value Decomposition, which is, for a matrix  $m \times n$ , an operation of time complexity  $O(m^2n + n^3)$ , with constant of proportionality ranging from 4 to 10 (or more) depending on the specific algorithm (Golub & Van Loan, 2013). But, as the dimension of the feature space does not grow larger, the dimension of the matrices stays constant even when more hand configurations are added. Therefore, the computing time grows only linearly when adding more configurations, as only linearly more distances have to be computed and compared.

Preliminary testing with people not involved in the development of the application has shown that they find it easy to use, with their performance on their past attempts helpful to let them know which configurations they find more challenging. For convenience we include in the repository the files needed to build a docker image, so the user might not need to install additional software.

The next step would be to recognize whole signs consisting of a succession of hand configurations in different spacial locations, as well as adding two-hands support and facial expressions. To this end, any approach suitable for time series could be applied, from HMMs to deep learning techniques.

#### CRediT authorship contribution statement

**Itsaso Rodríguez-Moreno:** Conceptualization, Software, Writing – original draft. **José María Martínez-Otzeta:** Methodology, Validation, Writing – review & editing, Supervision. **Basilio Sierra:** Validation, Writing – review & editing, Supervision.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgements

This work has been partially funded by the Basque Government, Spain, under Grant number IT1427-22; the Spanish Ministry of Science (MCIU), the State Research Agency (AEI), the European Regional Development Fund (FEDER), under Grant number PID2021-122402OB-C21 (MCIU/AEI/FEDER, UE); and the Spanish Ministry of Science, Innovation and Universities, under Grant FPU18/04737. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## References

- Ahmed, M. A., Zaidan, B. B., Zaidan, A. A., Salih, M. M., & Lakulu, M. M. (2018). A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. *Sensors*, *18*, 2208.
- Aktaş, M., Gökberk, B., & Akarun, L. (2019). Recognizing non-manual signs in Turkish sign language. *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, (pp. 1–6).
- Al-Qurishi, M., Khalid, T., & Souissi, R. (2021). *Deep learning for sign language recognition: Current techniques, benchmarks, and open issues*. IEEE Access.
- Azar, S. G., & Seyedarabi, H. (2020). Trajectory-based recognition of dynamic Persian sign language using hidden Markov model. *Computer Speech & Language*, *61*, Article 101053.
- Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, *37*, 1554–1563.
- Boháček, M., & Hrzů, M. (2022). Sign Pose-based Transformer for Word-level Sign Language Recognition. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, (pp. 182–191).
- Bradski, G. R. (1998). Real time face and object tracking as a component of a perceptual user interface. *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No. 98EX201)*, (pp. 214–219).
- Chen, X., Zou, D., & Su, F. (2021). Twenty-five years of computer-assisted language learning: A topic modeling analysis. *Language Learning & Technology*, *25*, 151–185.
- Cortes, C., & Vapnik, V. (1995). Support vector machine. *Machine Learning*, *20*, 273–297.
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, *1*, pp. 886–893.
- De Marco, R. M., & Foulds, R. A. (2003). Data recording and analysis of American Sign Language. *2003 IEEE 29th Annual Proceedings of Bioengineering Conference*, (pp. 49–50).
- Dryden, I. L., & Mardia, K. V. (2016). *Statistical shape analysis: with applications in R* (Vol. 995). John Wiley & Sons.
- Eberhard, D. M., Simons, G. F., & Fennig, C. D. (2022). *Ethnologue: Languages of the world*. *Ethnologue: Languages of the world*. Dallas, TX: SIL International.
- Fadlilah, U., Handaga, B., et al. (2021). The development of android for Indonesian sign language using tensorflow lite and CNN: An initial study. *Journal of Physics: Conference Series*, *1858*, Article 012085.
- Fazi, M. B. (2021). Beyond human: Deep learning, explainability and representation. *Theory, Culture & Society*, *38*, 55–77.
- Gao, L., Li, H., Liu, Z., Liu, Z., Wan, L., & Feng, W. (2021). RNN-transducer based Chinese sign language recognition. *Neurocomputing*, *434*, 45–54.
- Garcia, S., Derrac, J., Cano, J., & Herrera, F. (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*, 417–435.
- Golub, G. H., & Van Loan, C. F. (2013). *Matrix computations*. JHU press.
- Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, *40*, 33–51.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 770–778).
- Ibañez, R., Soria, Á., Teyseyre, A., Rodríguez, G., & Campo, M. (2017). Approximate string matching: A lightweight approach to recognize gestures with Kinect. *Pattern Recognition*, *62*, 73–86.
- Jiang, Y., Bosch, N., Baker, R. S., Paquette, L., Ocuppaugh, J., Andres, J. M., . . . Biswas, G. (2018). Expert feature-engineering vs. deep neural networks: which is better for sensor-free affect detection? *International conference on artificial intelligence in education*, (pp. 198–211).
- Mazoreanu, E. (2019). Market size of the global language services industry, 2009–2022. *Market size of the global language services industry, 2009–2022*.
- Quinn, M., & Olszewska, J. I. (2019). British sign language recognition in the wild based on multi-class SVM. *2019 federated conference on computer science and information systems (FedCSIS)*, (pp. 81–86).
- Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, *164*, Article 113794.
- Révy, G., Hadházi, D., & Hullám, G. (2022). Towards Hand-Over-Face Gesture Detection. *29th Minisymposium of the Department of Measurement and Information Systems*, (pp. 58–61).
- Rocha, J., Lensk, J., Ferreira, T., & Ferreira, M. (2020). Towards a tool to translate brazilian sign language (libras) to brazilian portuguese and improve communication with deaf. *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, (pp. 1–4).
- Roy, P. P., Kumar, P., & Kim, B.-G. (2021). An efficient sign language recognition (SLR) system using Camshift tracker and hidden Markov model (HMM). *SN Computer Science*, *2*, 1–15.
- Sevli, O., & Kemaloglu, N. (2020). Turkish sign language digits classification with CNN using different optimizers. *International Advanced Researches and Engineering Journal*, *4*, 200–207.
- Sharma, A., Sharma, N., Saxena, Y., Singh, A., & Sadhya, D. (2021). Benchmarking deep neural network approaches for Indian Sign Language recognition. *Neural Computing and Applications*, *33*, 6685–6696.
- Taskiran, M., Killioglu, M., & Kahraman, N. (2018). A real-time system for recognition of American sign language by using deep learning. *2018 41st international conference on telecommunications and signal processing (TSP)*, (pp. 1–5).
- Tremblay, J., Singh, K., Fern, A., Kirton, E. S., He, S., Woyke, T., . . . Tringe, S. G. (2015). Primer and platform effects on 16S rRNA tag sequencing. *Frontiers in microbiology*, *6*, 771.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.
- Wang, R. Y., & Popović, J. (2009). Real-time hand-tracking with a color glove. *ACM Transactions on Graphics (TOG)*, *28*, 1–8.
- Zabala, U., Rodríguez, I., Martínez-Otzeta, J. M., Irigoien, I., & Lazkano, E. (2021). Quantitative analysis of robot gesticulation behavior. *Autonomous Robots*, *45*, 175–189.
- Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., & Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*.
- Zhang, R., & Zou, D. (2022). Types, purposes, and effectiveness of state-of-the-art technologies for second and foreign language learning. *Computer Assisted Language Learning*, *35*, 696–742.

